

МИКРОПРОЦЕССОРНЫЕ СРЕДСТВА И СИСТЕМЫ

2 | 1986

ISSN 0233-4844

Микропроцессорный комплект БИС серии К1815 для конвейерной реализации алгоритмов преобразований Фурье, цифровой фильтрации, перемножения матриц, свертки и др., позволяющих решать задачи быстрой обработки сигналов

Модуль процессора персональной ЭВМ «Ириша»: принципиальная и монтажная схемы, топология печатной платы, описание режимов работы, состав программного обеспечения, устройства наладки

Программируемые логические матрицы (ПЛМ): процедуры программирования и контроля, технические средства записи данных, особенности применения в микропроцессорных системах

Персональная ЭВМ «Океан 240» на основе БИС серии КР580 отличается малым энергопотреблением (до 7 Вт) и может быть использована в измерительных комплексах с автономным питанием

Система команд микропроцессора КМ1810ВМ86: типы и структура команд, алгоритмы их выполнения, характеристика программно-доступных элементов, форматы команд и данных

Система программирования Диалог для микропроцессора К580ИК80 обеспечивает ввод и редактирование любого участка программы на уровне исходного текста, создание отладочных вариантов программы с быстрой загрузкой, генерацию загрузочного модуля





ПЕРСОНАЛЬНАЯ ЭВМ «ОКЕАН-240»

Одно из применений персональной ЭВМ «Океан-240» — экспрессобработка данных гидрологического зондирования.

(К ст. Тилинина Д. А.)



Одноплатная персональная ЭВМ «Океан 240», разработанная в Институте океанологии им. П. П. Ширшова АН СССР, предназначена для работы в условиях морской экспедиции. Выполнена на базе микропроцессорного комплекта БИС серии КР580, имеет ОЗУ емкостью 128К байт, ППЗУ 16К байт. Простота сопряжения с нестандартной измерительной аппаратурой обеспечивается набором устройств параллельного и последовательного обмена. Для отображения алфавитно-цифровой и графической информации используется цветной либо черно-белый ТВ-приемник, имеющий вход видеосигнала. Устройством внешней памяти служит бытовой кассетный магнитофон.

Персональная ЭВМ «Океан 240» отличается высокой надежностью, относительно низким энергопотреблением (около 7 Вт), малыми габаритами. Программное обеспечение ПЭВМ «Океан 240» состоит из резидентного монитора, обеспечивающего интерфейс с графическим дисплеем и устройствами ввода-вывода: последовательной линией RS232-C, печатающим устройством типа УВВПЧ-30-004, кассетным магнитофоном, алфавитно-цифровой клавиатурой и квазидисковой операционной системой, совместимой с ОС CP/M-80, использующей 64К байт ОЗУ в качестве «электронного диска».

Пролетарии всех стран, соединяйтесь!

ОРГАН ГОСУДАРСТВЕННОГО КОМИТЕТА СССР ПО НАУКЕ И ТЕХНИКЕ

Издается с 1984 года

МП

МИКРОПРОЦЕССОРНЫЕ СРЕДСТВА И СИСТЕМЫ

ВЫХОДИТ ШЕСТЬ РАЗ В ГОД НАУЧНО-ТЕХНИЧЕСКИЙ И ПРОИЗВОДСТВЕННЫЙ ЖУРНАЛ 2 / 1986 МОСКВА

СОДЕРЖАНИЕ

Адрес редакции: 101820, проезд Серова, 5, редакция журнала «Микропроцессорные средства и системы»
Телефоны 228-18-88; 221-99-26

Сдано в набор 03.03.86 Т 10131
Подписано к печати 15.04.86
Формат 84×108 1/16. Бумага № 1.
Уч. изд. л. 13,72. Тираж 52 900 экз.
Зак. № 47. Цена 1 р. 10 к.

Орган Государственного комитета СССР по науке и технике

Московская типография № 13
ПО «Периодика» ВО «Союзполиграфпром» Государственного комитета СССР по делам издательств, полиграфии и книжной торговли.
107005, Москва, Б-5, Денисовский пер., д. 30

На первой странице обложки — Диаграмма шахматной позиции (см. статью А. К. Дудоладова, с. 47)

ШКОЛЬНАЯ ИНФОРМАТИКА: ВТОРОЙ ЗВОНОК

В этом выпуске журнала публикуется перспективная программа 102-часового курса основ информатики и вычислительной техники для старших классов средней школы. Эта программа рекомендована Министерством просвещения СССР в качестве основы, своего рода «технического задания», на учебник, который должен появиться в результате объявленного открытого конкурса.

Трудно переоценить значимость этого документа. Пятьдесят лет назад невозможно было увидеть даже предпосылки к такому компоненту человеческой культуры. Двадцать пять лет назад эта сумма знаний еще только складывалась в фрагментах научно-исследовательской и преподавательской работы. Двенадцать лет назад аналогичный курс был доступен не любому вузу. А через десять лет мы будем свидетелями перенесения курса информатики в неполную среднюю шко-

Три главных умения должен выработать этот курс: во-первых — это прямое знакомство с компьютером, выработка позитивного и активного отношения к нему, преодоление психологического барьера, приобретение навыков свободной клавиатурной работы; во-вторых — это понимание того, что может и чего не может компьютер, способность поставить задачу для ЭВМ, найти алгоритм ее решения и выразить этот алгоритм на языке, понятном компьютеру; в третьих — это приобретение начального опыта практической работы на ЭВМ с использованием базового и прикладного программного обеспечения общего назначения.

При всей своей беспрецедентности и направленности в будущее, этот учебный план отличается взвешенностью и реализмом. Он существенно опирается на традиции отечественной системы образования и в то же время учитывает общемировую тенденцию перемен, вызванных научно-технической революцией и всеохватным распространением микропроцессорной вычислительной техники. Десятилетний опыт передовых научно-педагогических коллективов уже воспитал первые тысячи выпускников средней школы, владеющих основами информатики и вычислительной техники в объеме, близком к содержанию публикуемой программы.

И все же от реализма замысла до реальности воплощения курса школьной информатики — дистанция огромного размера, которая может быть преодолена только упорной и повседневной работой многих людей.

Самая главная задача — это фундаментализация «базы знаний» курса школьной информатики. Нужна неустанная работа мысли, чтобы вычленить из пестрого потока компьютерных приемов и терминов, правил и навыков зерна простых и глубоких понятий, базовую совокупность неделимых процедур, набор прямых наблюдений, опирающихся на живой опыт ребенка, систему рассуждения и стыковки элементарных навыков, позволяющую строить и обогащать интеллектуальный и операционный потенциал учащегося.

Затем этот понятийно-процедурный каркас информатики нужно одеть в одежды словесных объяснений, примеров и задач, соответствующих жизненному опыту подростков, реальностям школьного учебного процесса, законам психологии обучения и развития, а также возрастным особенностям.

Эти две проблемы являются общей научно-педагогической задачей, возникающей при любом охвате сферой образования новой науки и нового вида человеческой деятельности. Здесь же возникает третья, новая для педагогики задача разработки и введения в учебный процесс программного обеспечения курса школьной информатики. Это — тот участок работы, в котором помощь специалистов должна быть особенно широкой.

В исторических документах XXVII съезда нашей партии отмечено, что школьная реформа является лишь первым шагом в создании системы непрерывного образования. Это особенно касается информатики, как в ее теоретической, так и в прикладной частях. Компьютеризация является, одновременно, и стержнем и фокусом благодаря универсальности компьютера как орудия во всех сферах применения и как аккумулятора человеческого интеллекта в форме программного обеспечения.

Школьная информатика является дважды первенцем. ЭВМ в школе — это первое вступление к пожизненному партнерству с компьютером и это также первое массовое применение современной вычислительной техники. Эти два грандиозных дебюта выдвигают еще две новые проблемы.

Первая проблема — адаптация вузовского образования к новому образовательному уровню выпускников средней школы. Воздействие этого уровня начнет сказываться уже через несколько лет.

Вторая проблема — реализация, отработка нового образовательного уровня во всей практике применения ЭВМ. Понятийная и операционная база компьютерной грамотности должна будет найти свое отражение в упорядоченности профессиональной терминологии, в широком распространении методов систематического программирования, в унификации алгоритмической символики и, прежде всего, в существенно более интенсивной творческой работе по созданию отечественного программного обеспечения, технической, научной и учебной литературы. Здесь большая роль принадлежит научно-техническим и популярным журналам и другим средствам воздействия на человеческий фактор в ускоренном развитии нашего общества.

А. Ершов

УДК 681.322.068

А. В. Кобылинский, А. В. Береза, Н. Г. Сабадаш, А. К. Тесленко, В. А. Темченко

СИСТЕМА КОМАНД МИКРОПРОЦЕССОРА КМ1810ВМ86

Система команд микропроцессора КМ1810ВМ86 * позволяет повысить производительность микропроцессорных систем, сократить трудоемкость их программирования, обеспечить проектирование многопроцессорных систем и преемственность с широко используемыми системами на основе БИС серии КР580. Система команд этого микропроцессора содержит традиционные группы команд-пересылок (арифметические, поразрядной обработки данных, перехода и управления процессором) и группу команд обработки строк, обеспечивающей программирование и повышающей производительность обработки текстовых данных. Программно-доступные элементы адресуются с помощью неявной, непосредственной, прямой регистровой, косвенной регистровой и косвенной адресации.

Краткая характеристика программно-доступных элементов

В состав программно-доступных регистров микропроцессора КМ1810ВМ86 входят 16-разрядные регистры общего назначения (РОН) AX, BX, CX, DX, SP, BP, SI и DI, 16-разрядные сегментные регистры CS, DS, SS и ES, 16-разрядный регистр адреса команд IP и регистр признаков F. РОН AX, BX, CX и DX могут использоваться как восемь 8-разрядных регистров AL, BL, CL, DL, AH, BH, CH и DH.

Регистр F содержит признаки результата (CF — признак переноса, PF — четности, AF — вспомогательного переноса, ZF — нуля, SF — знака и OF — переполнения) и признаки управления (IF — признак разрешения прерывания, TF — шагового режима и DF — направления).

Общий объем адресуемой памяти микропроцессора КМ1810ВМ86 — 1 Мбайт. Структурные единицы памяти — байт, слово (2 байта), двойное слово и сегмент (64 байта). Адресом байта, слова и двойного слова может быть любое 20-разрядное двоичное число. Адрес сегмента памяти всегда содержит в четырех младших разрядах нули.

Двадцатиразрядный адрес структурных единиц памяти не является программно-доступным; старшие 16 разрядов адреса сегмента, называемые базой сегмента и располагаемые в сегментных регистрах, программно-доступны. Для адресации остальных структурных единиц памяти в программах используется логический адрес (указатель), содержащий две 16-разрядные компоненты: базу сегмента, помещаемую в сегментный регистр, и относительный адрес (смещение) в сегменте.

В один и тот же момент программе доступны четыре текущих сегмента по 64К байт каждый: сегмент кодов, адресуемый регистром CS; данных, адресуемый регистром DS; стека, адресуемый регистром SS, и вспомогательный сегмент, адресуемый регистром ES. Сегменты могут быть разнесены, перекрываться и совпадать. Объем и типы сегментов памяти обеспечивают преемственность с микропроцессором КР580ВМ80, в котором выделение сегментов кодов, данных и стека осуществляется на программном уровне.

Сегментная организация памяти обладает следующими достоинствами: сравнительная простота перемещения программ путем изменения содержимого сегментных регистров; понижение вероятности непредусмотренного алгоритмом разрушения команд и данных программ при разнесении сегментов в памяти; обеспечение дополнительных возможностей по организации мультипрограммных и мультипроцессорных систем с работой на общую память.

Адресация внешних устройств в микропроцессоре КМ1810ВМ86 осуществляется независимым полем адресов объемом 64К или через поле адресов памяти. Регистры внешних устройств могут быть 8- и 16-разрядные.

Форматы команд и данных

Команды микропроцессора КМ1810ВМ86 занимают в памяти от одного до шести байт. Первый байт команды содержит код операции. В некоторых командах код операции размещается и во втором байте. На рисунке



Типичный формат команд микропроцессора КМ1810ВМ86

приведен обобщенный формат типичной команды микропроцессора КМ1810ВМ86. Одноразрядное поле W имеется в подавляющем большинстве команд и определяет разрядность операндов. Если W = 0, то операнды 8-разрядные; если W = 1, то 16-разрядные. Поля mod и r/m второго байта команды (табл. 1) определяют один из 24 режимов формирования относительного адреса операнда в сегменте данных или в сегменте стека. Поля disp8 и disp16 определяют одно- или двухбайтное смещение в команде. Если смещение однобайтное, то производится его знаковое расширение до двух байт (т. е. значение всех разрядов второго байта устанавливается равным значению старшего разряда исходного байта).

Таким образом, микропроцессор КМ1810ВМ86 обеспечивает однокомпонентную прямую и косвенную адресацию элементарных данных, многовариантную двухкомпонентную адресацию, удобную, например, для доступа к элементам массивов и простых структур, и трехкомпонентную адресацию, удобную для доступа к элементам сложных структур данных, используемых в языках высокого уровня.

---

* Описание схемотехнических особенностей реализации и структуры микропроцессора, включая краткий обзор системы команд и методов адресации команд и данных, опубликовано в журнале «Микропроцессорные средства и системы», 1986, № 1, с. 28—33.

Таблица 1

Режимы формирования относительного адреса (смещения в сегменте)

r/m	mod					
	00		01		10	
	Смещение	Сегмент	Смещение	Сегмент	Смещение	Сегмент
000	BX+SI	Данных	BX+SI+disp 8	Данных	BX+SI+disp 16	Данных
001	BX+DI	»	BX+DI+disp 8	»	BX+DI+disp 16	»
010	BP+SI	Стека	BP+SI+disp 8	Стека	BP+SI+disp 16	Стека
011	BP+DI	»	BP+DI+disp 8	»	BP+DI+disp 16	»
100	SI	Данных	SI+disp 8	Данных	SI+disp 16	Данных
101	DI	»	DI+disp 8	»	DI+disp 16	»
110	disp 16	»	BP+disp 8	Стека	BP+disp 16	Стека
111	BX	»	BX+disp 8	Данных	BX+disp 16	Данных

Поле r/m при mod=11, а также поле reg определяют РОН (табл. 2).

Поля data8 или data16 определяют одно- или двухбайтные непосредственные данные.

Таблица 2

Адресация РОН

Поле r/m или reg	W=0	W=1	Поле r/m или reg	W=0	W=1
000	AL	AX	100	AH	SP
001	CL	CX	101	CH	BP
010	DL	DX	110	DH	SI
011	BL	BX	111	BH	DI

Приведенный формат типичных команд отражает их симметричность при использовании операндов. Действительно, при одном, и том же коде операции, меняя значения полей mod, reg и r/m, можно в качестве операндов использовать содержимое любого из РОН или памяти. Однако в формировании исполнительного адреса (смещения в сегменте) каждый из регистров в микропроцессоре КМ1810ВМ86 имеет индивидуальное назначение. Это способствует сокращению потенциальных ошибок в выполнении соглашений о связях между программами, где даже в случае универсальности регистров большинство из них имеют специальное назначение.

Кроме типичного формата, в микропроцессоре КМ1810ВМ86 реализована группа команд с упрощенным форматом, которые осуществляют наиболее часто применяемые операции. По быстродействию и занимаемой памяти они более эффективны. Так, команды упрощенного формата для работы с РОН являются однобайтными, а поле reg помещено в байт с кодом операции. При работе с данными в памяти используется регистр AX, а данные в сегменте данных адресуются вторым и третьим байтами команды.

Микропроцессор КМ1810ВМ86 обеспечивает аппаратную обработку элементарных и индексированных данных, стека, строки и таблицы. К элементарным данным относятся одно- и двухбайтные порядковые и целые числа, однобайтные упакованные, одно- и двухбайтные распакованные двоично-десятичные числа, а также четырехбайтные логические адреса (указатели).

Целые числа представляются в дополнительном коде. Реализация в микропроцессоре признака переполнения OF практически позволяет использовать модифицированный дополнительный код. В связи с этим диапазон изменения целых однобайтных чисел от −255 до +255, двухбайтных от −65 535 до +65 535.

Диапазон изменения однобайтных порядковых чисел 0...255, двухбайтных 0...65 535.

Упакованные двоично-десятичные числа содержат две десятичные цифры в байте, распакованные — одну.

Указатель в младшем слове содержит смещение в сегменте, в старшем — базу сегмента; слово в младшем байте — младшую часть значения, в старшем — старшую часть.

Над порядковыми, целыми и распакованными двоично-десятичными числами выполняются операции сложения, вычитания, умножения и деления. Над упакованными двоично-десятичными числами — операции сложения и вычитания.

К индексированным данным относятся массивы и структуры, при обращении к элементам которых используются отмеченные ранее режимы адресации (см. табл. 1).

В микропроцессоре КМ1810ВМ86 под стек отводится специальный сегмент, адресуемый сегментным регистром SS. Указатель стека, определяющий относительный адрес верхушки стека в сегменте, располагается в регистре SP. При записи в стек вначале содержимое регистра SP уменьшается на два, а затем по адресу в SP записывается слово данных. При чтении вначале по адресу, находящемуся в SP, читается слово данных, затем значение SP увеличивается на два. Реализованы команды обмена данными между стеком и регистрами микропроцессора, включая сегментные регистры, а также, в отличие от микропроцессора КР580ИК80, между стеком и памятью. Кроме хранения адресов возврата из подпрограмм (что обеспечивается аппаратно), стек наиболее часто служит для передачи фактических параметров. При засылке содержимого SP в BP для доступа к фактическим параметрам используются различные режимы адресации (см. табл. 1). Это позволяет повысить эффективность программ. Строковые данные и таблицы обрабатываются отдельными командами.

Команды микропроцессора КМ1810ВМ86 описываются в соответствии с их классификацией по функциональным группам. Обозначения и алгоритмы выполнения команд сведены в табл. 3. Если в обозначении операнда разрядность не указывается, то в зависимости от значения поля W команда может обрабатывать 8- или 16-разрядные операнды.

Система команд

Команды пересылок (общего назначения, пересылки адресов, пересылки признаков, ввода-вывода и перекодировки) обеспечивают передачу операнда-источника в операнд-приемника без содержательного их преобразования (табл. 4). Команды пересылок общего назначения подразделяются на симметричные команды для указания операндов поля mod, r/m и reg во втором байте и команды с упрощенным форматом.

Таблица 3

Обозначения, используемые в описании алгоритмов

Обозначение	Характеристика
E	Операнд в памяти или в РОН, определяемый полями mod и r/m
R	Операнд в РОН
Rseg	Операнд в сегментном регистре
D	Непосредственные данные в команде
A	Если W=0, то операнд в регистре AL. Если W=1, то операнд в регистре AX
St	Стек
M(...)	Операнд в памяти, в скобках указывается смещение в сегменте
Port(...)	Операнд в регистре внешнего устройства, в скобках указывается адрес устройства
EA	Исполнительный адрес операнда в памяти, определяемый полями mod и r/m
ext	Знаковое расширение байта в словах или слова в двойное слово
&	Объединение байт в слово, слов — в двойное слово
B2 и B3	Второй и третий байт команды
f(...)	Функция определения значений признаков результата по результату операции, указанной в скобках

Таблица 4

Команды пересылок

Мнемоника	Алгоритмы
	Команды общего назначения
MOV	E ← R, R ← E, E ← D
PUSH; POP	St ← E16; E16 ← St
XCHG	E ↔ R
	Команды общего назначения упрощенного формата
MOV	R ← D, A ← M(B3 & B2), M(B3 & B2) ← A
PUSH; POP	St ← R16; R16 ← St
XCHG	AX ↔ R16
	Команды пересылки адресов
MOV	Rseg←E(Rseg≠CS), E←Rseg
PUSH; POP	St←Rseg; Rseg←St(Rseg≠CS)
LEA	R16 ← EA
LDS	DS ← M16(EA+2), R16 ← M16(EA)
LES	ES ← M16(EA+2), R16 ← M16(EA)
	Команды пересылки признаков
PUSHF; POPF	St ← F; F ← St
LAHF; SAHF	AH ← F; F ← AH
	Команды ввода-вывода
IN	A ← Port(B2), A ← Port(DX)
OUT	Port(B2) ← A, Port(DX) ← A
	Команды перекодировки
XLAT	AL ← M(BX+AL)

Среди команд с упрощенным форматом имеется команда XCHG AX, AX, которая используется как команда NOP (нет операции). Специальная команда NOP в микропроцессоре КМ1810ВМ86 отсутствует.

Команды пересылок адресов позволяют динамично изменять операционную обстановку, т. е. определять текущие сегменты памяти. Следует отметить, что после команд, изменяющих содержимое сегментного регистра (MOV и POP), запрещаются прерывания на время выполнения следующей команды. В качестве следующей команды рекомендуется использовать команду пересылки в один из регистров компоненты смещения логического адреса. Это обеспечивает полную загрузку логических адресов (указателей), что особенно важно при загрузке регистров SS и SP. Команда LEA осуществляет программную доступность исполнительного адреса. Она позволяет организовать доступ к элементам сложных структур данных (например, многомерных массивов). Команды LDS и LES обеспечивают загрузку указателей и особенно эффективны для подготовки обработки строковых данных.

Адресация регистров внешних устройств в командах **ввода-вывода** может быть прямая (в пределах первых 256 регистров) и косвенная через регистр DX. Тем самым, с одной стороны, достигается преемственность с микропроцессором КР580ИК80, а с другой благодаря косвенной адресации повышается гибкость адресации внешних устройств.

Команда перекодировки XLAT позволяет обрабатывать таблицы размером 256 байт. Начало таблицы в сегменте данных указывается в регистре BX. Команда выполняет пересылку в AL байта таблицы, находящегося по номеру в регистре AL. При соответствующей подготовке таблицы обеспечивается преобразование символьных строк, например, с кода КОИ-7 в код ДКОИ и т. д.

Арифметические команды (сложения, вычитания, сложения и вычитания упрощенного формата, умножения, деления и преобразования форматов данных) предназначены для выполнения основных арифметических операций над порядковыми, целыми и двоично-десятичными числами. Алгоритмы команд представлены в табл. 5.

Все команды вычитания и сложения (за исключением команд коррекции) устанавливают все признаки результата. Команды INC и DEC признак переноса оставляют без изменения. Команды ADD и SBB обеспечивают выполнение сложения и вычитания многобайтовых чисел. Порядковые, целые и двоично-десятичные числа обрабатываются одними и теми же командами сложения и вычитания.

Для получения правильного результата для двоично-десятичных чисел необходимо произвести коррекцию. В случае сложения этим целям для распакованных чисел служит команда AAA, для упакованных — команда DAA. Коррекция происходит, если в результате двоичного сложения в тетраде образовалось число больше 9 или возник перенос в следующую тетраду (т. е. результат сложения двух десятичных цифр больше 16). В первом случае необходимо вычесть из тетрады 10, во втором — прибавить в тетраду 6. Поскольку вычитание из тетрады равносильно прибавлению 6, то в обоих случаях для коррекции результата прибавляют 6. При коррекции распакованных чисел обрабатывается только младшая тетрада, а старшая тетрада обнуляется. Корректируемый байт располагается в регистре AL. Если была необходимость прибавления 6, то это свидетельствует о переносе в старший десятичный разряд, поэтому признак CF устанавливается в 1. Кроме того, к регистру AH добавляется 1. Тем самым повышаются удобства обработки многоразрядных двоично-десятичных чисел.

В случае упакованных двоично-десятичных чисел вначале согласно рассмотренному способу обрабатывается младшая тетрада. Если после прибавления числа 6 возникает перенос, то он складывается со старшей тетрадой, которая затем обрабатывается.

Для коррекции результата вычитания распакованных двоично-десятичных чисел используется команда AAS упакованных DAS. Отличие алгоритма коррекции вы-

Таблица 5

Арифметические команды

Мнемоника	Алгоритмы
	Команды сложения
ADD	$E \leftarrow E+R$, $R \leftarrow R+E$, $E \leftarrow E+D$, $E \leftarrow E+Dext$
ADC	$E \leftarrow E+CF+R$, $R \leftarrow R+CF+E$, $E \leftarrow E+CF+D$, $E \leftarrow E+CF+Dext$
INC	$E \leftarrow E+1$
AAA, DAA	Коррекция результата сложения распакованных и упакованных двоично-десятичных чисел
	Команды вычитания
SUB	$E \leftarrow E-R$, $R \leftarrow R-E$, $E \leftarrow E-D$, $E \leftarrow E-Dext$
SBB	$E \leftarrow E-CF-R$, $R \leftarrow R-CF-E$, $E \leftarrow E-CF-D$, $E \leftarrow E-CF-Dext$
DEC	$E \leftarrow E-1$
NEG	$E \leftarrow 0-E$
CMP	$F \leftarrow f(E-R)$, $F \leftarrow f(R-E)$, $F \leftarrow f(E-D)$, $F \leftarrow f(E-Dext)$
AAS, DAS	Коррекция результата вычитания
	Команды сложения и вычитания упрощенного формата
ADD, ADC	$A \leftarrow A+D$, $A \leftarrow A+CF+D$
INC, DEC	$R \leftarrow R+1$, $R \leftarrow R-1$
CUB, SBB	$A \leftarrow A-D$, $A \leftarrow A-CF-D$
CMP	$F \leftarrow f(A-D)$
	Команды умножения
MUL; IMUL	$\underbrace{AX \leftarrow AL \times E}_{W=0}$; $\underbrace{DX\ \&\ AX \leftarrow AX \times E}_{W=1}$
DIV; IDIV	$AL \leftarrow AX/E$, $AH \leftarrow mod(AX, E)$ } $W=0$; $AX \leftarrow (DX\ \&\ AX)/E$, $DX \leftarrow mod((DX\ \&\ AX), E$ } $W=1$
	Команды преобразования форматов данных
AAM	Преобразование двоичного числа в регистре AL в распакованное двоично-десятичное число в регистрах AL и AH
AAD	Преобразование распакованного двоично-десятичного числа в AH и AL в двоичное число в AL
CBW	$AH \leftarrow ALext$
CWD	$DX\&AX \leftarrow AXext$

читания состоит в том, что при наличии заема со старшей тетрады из текущей тетрады необходимо вычесть 6, так как двоичный заем равен 16, а не 10.

Удобство обработки многоразрядных двоично-десятичных чисел обеспечивается установкой признака переноса, как и при сложении.

Команды CMP, служащие для сравнения данных, работают так же, как команды SUB, однако результат в операнд-приемник не поступает, изменяется только состояние признаков результата.

В группу команд умножения входят команды MUL для умножения порядковых чисел и IMUL для умножения целых чисел (со знаком). Для однобайтных данных один из сомножителей должен быть в AL, результат помещается в регистр AX. Для двухбайтных данных один из сомножителей помещается в регистр AX, младшее слово результата помещается в AX, старшее — в DX. После выполнения команд умножения устанавливаются признаки переноса и переполнения.

В группу команд деления входят команды DIV для деления порядковых чисел и IDIV для деления целых чисел (со знаком). Делимое занимает регистр AX при W=0 или регистры DX (старшее слово) и AX при W=1. Целая часть частного помещается в регистр AH (AX при W=1), остаток — в регистр AH (DH при W=1). Если при делении порядковых чисел частное больше 255 (65 535 при W=1) или при делении целых чисел модуль частного больше 127 (32 767 при W=1), или делитель равен 0, то возникает внутреннее прерывание по ошибке деления. После выполнения команды деления признаки неопределенны.

Команды преобразования байта в слово CBW и слова в двойное слово CWD обеспечивают знаковое расширение байта и слова соответственно.

Команда AAM используется для коррекции результата двоичного умножения двух однобайтных распакованных двоично-десятичных чисел; команда AAD — для подготовки к делению двухбайтного распакованного двоично-десятичного числа на однобайтное.

К командам поразрядной обработки данных относятся команды логические линейного сдвига и команды циклического сдвига (табл. 6). Логические команды реализуют стандартный набор поразрядных логических операций НЕ ($y=\bar{x}$), И ($z=x \wedge y$), ИЛИ ($z=x \vee y$) и исключающее ИЛИ (неравнозначность, сумма по mod2), обозначаемую $z=x \oplus y$. В зависимости от значения поля W операнды логических команд могут быть одно- или двухбайтными. Все команды, за исключением NOT, устанавливают признаки OF и CF в состояние «0», признаки SF, ZF и PF — по результату. Значение признака AF определенно. Команда NOT

Таблица 6

Команды поразрядной обработки данных

Мнемоника	Алгоритмы
	Логические команды
NOT	$E \leftarrow \bar{E}$
AND	$E \leftarrow E \wedge R$, $R \leftarrow R \wedge E$, $E \leftarrow E \wedge D$
OR	$E \leftarrow E \vee R$, $R \leftarrow R \vee E$, $E \leftarrow E \vee D$
XOR	$E \leftarrow E \oplus R$, $R \leftarrow R \oplus E$, $E \leftarrow E \oplus D$
TEST	$F \leftarrow f(E \wedge R)$, $F \leftarrow f(R \wedge E)$, $F \leftarrow f(E \wedge D)$
	Логические команды упрощенного формата
AND:OR	$A \leftarrow A \wedge D : A \leftarrow A \vee D$
XOR:TEST	$A \leftarrow A \oplus D : F \leftarrow f(A \wedge D)$
	Команды линейного сдвига
SHL	$E \leftarrow E \times 2$, $E \leftarrow E \times 2^{CX}$
SAR	$E \leftarrow E/2$, $E \leftarrow E/2^{CX}$, / — деление целых чисел
SHR	$E \leftarrow E/2$, $E \leftarrow E/2^{CX}$, / — деление порядковых чисел
	Команды циклического сдвига
ROL	1 или CL раз; CF; E
ROR	1 или CL раз; CF; E
RCL	1 или CL раз; CF; E
RCR	1 или CL раз; CF; E

не изменяет признаки. Команда TEST выполняет поразрядную логическую операцию И над исходными операндами, устанавливает признаки, но результат в операнд-приемник не отсылает.

В отличие от микропроцессора КР580ИК80 в КМ1810ВМ86 реализованы операции циклического и линейного сдвига на один разряд и на программно устанавливаемое в регистре CL число разрядов.

При выполнении линейного сдвига влево (команда SHL) признак переноса устанавливается по значению, выдвигаемому из старшего разряда. После этого, если их значения не равны, в состояние «1» устанавливается признак переполнения. Следовательно, команду SHL порядковых чисел можно использовать для сдвига влево и целых чисел. При сдвиге вправо признак переноса устанавливается в состояние, соответствующее биту, выдвигаемому из младшего разряда. Старшие разряды по команде SHR заполняются нулями (логический сдвиг), а по команде SAR — значением старшего (знакового) разряда (арифметический сдвиг).

Строковые команды (MOVS, CMPS, SCAS, LODS и STOS) позволяют не только вести расчеты, но и обрабатывать тексты. Обработка строковых данных в микропроцессоре КМ1810ВМ86 достигается с помощью однобайтных команд для типичных операций над отдельными элементами строк и аппаратной реализации многократного выполнения таких команд путем задания специального однобайтного префикса. В качестве элементов строк могут выступать одно- или двухбайтные данные. Строки-источники располагаются в сегменте данных, смещение элементов строк в сегменте задается содержимым регистра SI. Строки-приемники располагаются во вспомогательном сегменте, смещение элементов строк задается содержимым регистра DI. Строковые команды автоматически увеличивают (автоинкремент) или уменьшают (автодекремент) содержимое регистров SI и DI на единицу или на два. Автоинкремент или автодекремент задается признаком направления DF.

Команда MOVS выполняет пересылку элемента строки-источника по адресу элемента строки-приемника; CMPS путем вычитания элемента строки-приемника из элемента строки-источника сравнивает их (устанавливает признаки результата); LODS загружает элемент строки источника в регистр AX; SCAS сравнивает содержимое регистра AX и элемента строки-приемника; STOS пересылает содержимое регистра AX по адресу элемента строки-приемника.

Для выполнения действий над строками в программе непосредственно перед строковыми командами необходимо размещать однобайтный префикс повторения. Он задает многократное выполнение последующей строковой команды, сопровождаемое вычитанием единицы из регистра CX. Перед каждым выполнением строковой команды проверяется содержимое регистра CX и, если оно равно нулю, повторение выполнения прекращается.

Для строковых команд (CMPS и SCAX) условием выхода из аппаратного цикла является $CX=0 \vee ZF=z$, где z — значение младшего разряда байта префикса повторения.

Таким образом, строковые команды позволяют пересылать и сравнивать строки, производить поиск элемента в строке (или пропуск некоторого элемента в строке) и заполнять строки конкретным символом (например, пробелом). Строковые команды можно использовать и для обработки массивов данных других типов (например, целых, порядковых и т. п.).

Команды передачи управления (безусловной передачи управления, организации подпрограмм, прерывания, условного перехода и организации циклов) предназначены для изменения естественного порядка выполнения программы (табл. 7).

Команды JMP и CALL организуют переходы как внутри текущего сегмента кодов, так и на другой сегмент кодов, который становится текущим. Кроме прерывания, только эти команды позволяют переходить из

Таблица 7

Команды передачи управления

Мнемоника	Алгоритм
	Команды безусловной передачи управления
JMP	Межсегментная прямая IP ← B3 & B2 CS ← B5 & B4 Межсегментная косвенная CS & IP ← M16(EA+2) & M16(EA) Внутрисегментная прямая IP ← IP+B3 & B2 Внутрисегментная прямая короткая IP ← IP+B2 ext Внутрисегментная косвенная IP ← E
	Команды организации подпрограмм
CALL	Межсегментная прямая St ← CS St ← IP IP ← B3 & B2 CS ← B5 & B4 Межсегментная косвенная St ← CS St ← IP CS & IP ← M16(EA+2) & M16(EA) Внутрисегментная прямая St ← IP IP ← IP+B3 & B2 Внутрисегментная косвенная St ← IP IP ← E
RET	Межсегментная IP ← St CS ← St Внутрисегментная IP ← St
	Команды прерывания
INT	St ← F St ← CS St ← IP CS & IP ← Ptr(B2) IF ← 0 TF ← 0 Отличие от INT:
INT 3	CS & IP ← Ptr(3) Отличие от INT:
INT 0	если OF=1, то CS & IP ← Ptr(4)
IRET	IP ← St CS ← St F ← St

сегмента в сегмент. При естественном порядке выполнения команд по достижении конца сегмента управление автоматически передается на начало сегмента.

Команды RET подразделяются на команды с принудительным приращением указателя стека и без такового. Приращение задается во втором и третьем байтах команды и складывается с содержимым SP. Это восстанавливает исходное состояние указателя стека при возврате из процедур без выталкивания фактических параметров.

К командам прерывания относятся: INT — двухбайтная, задающая программное прерывание, определяемое пользователем; INT3 и INT0 — однобайтные, определяющие программные прерывания по точкам разрыва и по переполнению. Указатель для загрузки регистров CS и IP при выполнении этих команд выбирается из одной 256 четырехбайтных областей в начале поля адресов. Указатель (см. табл. 7) обозначается Ptr, а в скобках записывается номер четырехбайтной области. Команды INT используются, например, для упрощения программного интерфейса сложных программных комплексов — эмуляторов, операционных систем и др.; INT3 удобна в отладчиках; INT0 — для контроля переполнения.

Команда IRET служит для возврата из процедур обработки программных или аппаратных прерываний.

Микропроцессор КМ1810ВМ86 реализует разнообразные команды условного перехода. Все они

Таблица 8

Команды условного перехода

Мнемокод	Отношение	Тип	Условие
JE JZ	$a=b$ $a=0$	Целые, порядковые —	$ZF=1$
JNE JNZ	$a\neq b$ $a\neq 0$	Целые, порядковые —	$ZF=0$
JB JNAE JC	$a<b$ $(a\geqslant b)$ —	Порядковые » —	$CF=1$
JNB JAE JNC	$a>b$ $(a\leqslant b)$ —	Порядковые » —	$CF=0$
JG JNLE	$a>b$ $(a\leqslant b)$	Целые »	$((SF\oplus OF)\vee ZF)=0$
JGE JNL	$a\geqslant b$ $(a<b)$	Целые »	$(SF\oplus OF)=0$
JL JNGE	$a<b$ $(a\geqslant b)$	Целые »	$(SF\oplus OF)=1$
JLE JNG	$a\leqslant b$ $(a>b)$	Целые »	$((SF\oplus OF)\vee ZF)=1$
JA JNBE	$a>b$ $(a\leqslant b)$	Порядковые »	$(CF\vee ZF)=0$
JBE JNA	$a\leqslant b$ $(l>b)$	Порядковые »	$(CF\vee ZF)=1$
JO	—	—	$OF=1$
JNO	—	—	$OF=0$
JS	—	—	$SF=1$
JNS	—	—	$SF=0$
JP, JPE	—	—	$PF=1$
JNP, JPO	—	—	$PF=0$
JCXZ	—	—	$CX=0$

Таблица 9

Команды управления признаками состояния

Мнемоника	Алгоритм
STC, CLC, CMC	$CF\leftarrow 1,\ CF\leftarrow 0,\ CF\leftarrow \overline{CF}$
STD, CLD	$DF\leftarrow 1,\ DF\leftarrow 0$
STI, CLI	$IF\leftarrow 1,\ IF\leftarrow 0$

являются двухбайтными, причем при выполнении условия второй байт команды знаково расширяется до слова и складывается с содержимым регистра IP. Тем самым обеспечивается условный переход в пределах +127...—128 байт от адреса первого байта следующей команды. Анализируемые условия и мнемокоды команд перехода по условию представлены в табл. 8. Для улучшения выразительности программ для многих команд условного перехода задано несколько мнемокодов, каждый из которых выбирается в зависимости от контекста и позволяет акцентировать внимание на сущность обрабатываемых данных и особенности алгоритма.

LOOP, LOOPZ и LOOPNZ — команды организации циклов. Выполнение этих команд заключается в уменьшении индекса цикла, помещаемого в регистр CX, а при выполнении условия к содержимому IP прибавляется значение второго байта команды, знаково расширенное до слова. В команде LOOP условием перехода является $CX\neq 0$, в команде LOOPS — $CX\neq 0 \wedge ZF=1$, а в команде LOOPNZ — $CX\neq 0 \wedge ZF=0$.

Команды управления микропроцессором подразделяются на команды управления признаками состояния (табл. 9) и синхронизации (WAIT, ESC и HLT).

Команда WAIT с пятитактным интервалом анализирует состояние вывода TEST микропроцессора. Переход на выполнение следующей команды происходит лишь после того, как сигнал на выводе $\overline{TEST}$ станет равным 0. Это позволяет синхронизовать выполнение программы с некоторым внешним процессом.

Команда ESC организует расширения системы команд с помощью некоторого сопроцессора. По содержимому полей mod и r/m, а также сегментного регистра формируется полный физический адрес данных в памяти и выдается на шину адреса. Тем самым сопроцессор может использовать режимы адресации центрального процессора.

Команда HLT переводит микропроцессор в режим останова, из которого его можно вывести прерыванием или начальной установкой и запуском. По начальной установке все разряды регистра сегмента команд устанавливаются в состояние «1», а указатель команд, регистр признаков и остальные сегментные регистры в состояние «0».

Кроме префикса повторения строковых команд микропроцессора, КМ1810ВМ86 может обрабатывать еще два однобайтных префикса: блокировки шины LOCK и замены сегмента. Префикс LOCK обеспечивает организацию монопольного доступа микропроцессора к шине на все время выполнения следующей за ним команды.

Префикс замены сегмента позволяет отменить обычное использование сегмента и на время выполнения последующей команды принудительно установить требуе-

Таблица 10

Использование сегментов

Данные	Сегмент			
	Кодов (CS)	Данных (DS)	Стека (SS)	Вспомогательный (ES)
Коды команд	+	—	—	—
Элементарные данные	S	+	S	S
Данные, индексированные посредством регистра BP	S	S	+	S
Другие индексированные данные	S	+	S	S
Стек	—	—	+	—
Строка-источник	S	+	S	S
Строка-приемник	—	—	—	+
Таблица	S	+	S	S

мый сегмент. В табл. 10 показаны сегменты по умолчанию (+) и сегменты с префиксом (S). Префикс замены сегмента предотвращает многократные изменения содержимого сегментных регистров.

Рассмотренная система команд микропроцессора КМ1810ВМ86 по функциональной сложности реализованных операций, разнообразию способов адресации программно-доступных элементов, шинам и форматам обрабатываемых данных соответствует системам команд современных мини-ЭВМ, например СМ-4. В то же время она отражает многие современные тенденции в области реализации вычислительных систем и программирования и в этом смысле является более эффективной и перспективной. Это касается сегментации памяти, обеспечения взаимодействия с другими процессорами в многопроцессорной системе, более мощной поддержке аппарата процедур, сочетания симметричности РОН с их специализацией, особенно для целей адресации, вычисления многокомпонентного исполнительного адреса для упрощения доступа к сложным структурам данных, используемых в языках высокого уровня, и, наконец реализации операций обработки строковых данных.

Статья поступила 27 февраля 1986 г.

УДК 621.3.049.77 : 681.325.5

А. И. Белоус, В. В. Горовой, А. В. Прибыльский, А. В. Силин

ИНТЕРФЕЙСНЫЕ ИНТЕГРАЛЬНЫЕ МИКРОСХЕМЫ СЕРИИ К583

Интерфейсные интегральные микросхемы серии К583 — магистральный приемопередатчик с памятью К583ВА1, магистральный приемопередатчик К583ВА2, универсальный 8-разрядный коммутатор магистралей К583ВА3, универсальный 8-разрядный магистральный коммутатор К583ВА4 — выполняют обмен данными, логико-арифметическую обработку информации в современных вычислительных устройствах и применимы для сопряжения блоков вычислительных устройств с внешними устройствами. Они характеризуются широтой выполняемых функций, высоким быстродействием, большой нагрузочной способностью и низким уровнем рассеиваемой мощности.

Микросхемы выполнены на маломощных логических ТТЛШ схемах по базовой планарно-эпитаксиальной технологии с использованием окисной изоляции и двухуровневой металлизации.

Микросхема К583ВА1 (рис. 1) — магистральный приемопередатчик (МПП) с памятью — предназначена для согласования и обмена информацией между двумя типовыми магистралями и мощной магистралью, применяется в качестве мультиплексора, коммутатора, буферного регистра, усилителя мощности, контрольного устройства по проверке и формированию контрольных кодов передаваемой информации.

Микросхема обеспечивает (рис. 2) стандартные ТТЛ-уровни выходных-входных сигналов для магистралей L1, L2 (см. ниже). Временные диаграммы работы микросхемы приведены на рис. 3, *а*, *б*.

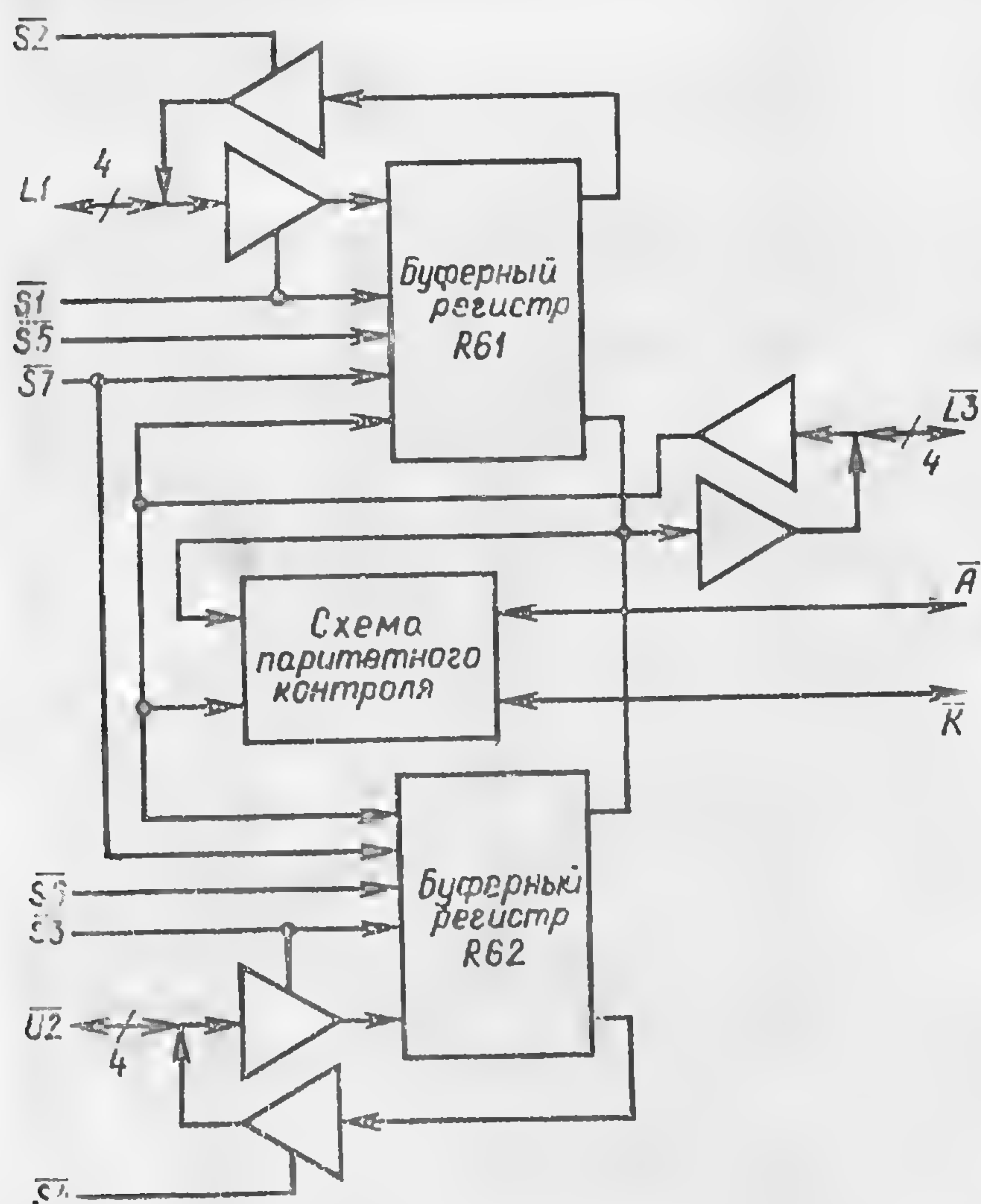


Рис. 1. Структурная схема микросхемы К583ВА1

Выполняемая операция	Синхросигналы						
	S1	S2	S3	S4	S5	S6	S7
L1→P1, L2→P2	⊔	X	⊔	X	1	1	0
L1→P1	⊔	X	1	X	1	1	0
L2→P2	1	X	⊔	X	1	1	0
Нет записи с L1 и L2	1	X	1	X	1	1	0
(К) L1→R1→$\overline{L3}$	⊔	X	1	X	0	1	0
(К) L2→R2→$\overline{L3}$	1	X	⊔	X	1	0	0
(К) L1→R1→$\overline{L3}$, L2→R2	⊔	X	⊔	X	0	1	0
(К) L2→R2→$\overline{L3}$, L1→R1	⊔	X	⊔	X	1	0	0
V(L1,L2)→V(R1,R2)→$\overline{L3}$	⊔	X	⊔	X	0	0	0
(К) R1→$\overline{L3}$	1	X	1	X	0	1	0
(К) R2→$\overline{L3}$	1	X	1	X	1	0	0
(Л) $\overline{L3}$→R1	X	1	X	1	⊔	1	1
(Л) $\overline{L3}$→R2	X	1	X	1	1	⊔	1
(Л) $\overline{L3}$→R2, $\overline{L3}$→R1	X	1	X	1	⊔	⊔	1
(П) $\overline{L3}$→R1→L1	X	0	X	1	⊔	1	1
(А) $\overline{L3}$→R2→L2	X	1	X	0	1	⊔	1
Нет записи с $\overline{L3}$	X	1	X	1	1	1	1
$\overline{L3}$→R1→L1, $\overline{L3}$→R2→L2	X	0	X	0	⊔	⊔	1
R1→L1	X	0	X	1	1	1	1
R2→L2	X	1	X	0	1	1	1

Примечание. В скобках указаны выводы, по которым осуществляется паритетный контроль.

Рис. 2. Алгоритм функционирования микросхемы К583ВА1

Основные электрические параметры микросхемы К583ВА1

Напряжение питания U_{cc}, В	5±10%
Ток потребления I_{cc}, мА, не более	100
Выходной ток низкого уровня I_{IL}, мА, не более:	
по входам $\overline{S5}$, $\overline{S6}$, $\overline{S7}$	|−0,5|
по входам $\overline{S2}$, $\overline{S3}$, $\overline{S4}$, $\overline{S1}$, $\overline{L1}$, $\overline{L2}$	|−0,25|
по входам $\overline{L3}$, $\overline{A}$, $\overline{K}$	|−0,2|
Входной ток высокого уровня I_{IH}, мкА, не более	50
Выходное напряжение высокого уровня U_{OH}, В, не менее (по магистрали $\overline{L3}$)	2,4
Выходной ток низкого уровня I_{OL}, мА, не менее (по магистрали L1, L2)	20
Время задержки передачи информации, t_D, нс, не более	110

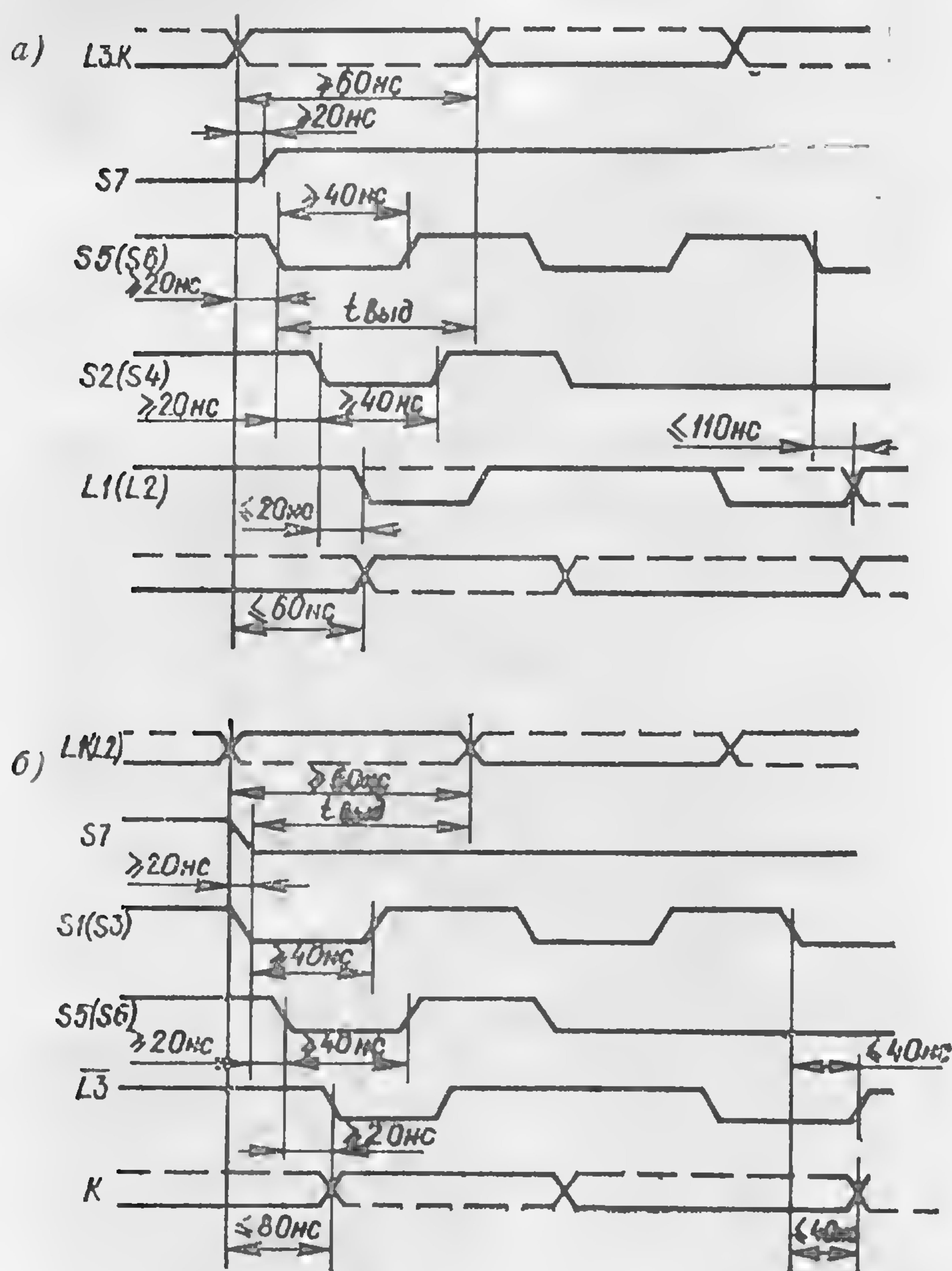


Рис. 3. Временные диаграммы работы микросхемы К583ВА1

Конструктивно микросхема выполнена в стандартном 24-выводном металлокерамическом корпусе.

Магистральный приемопередатчик (МПП) с памятью выполняет:

межмагистральный обмен данными между мощной магистралью и двумя типовыми магистралями в четырех направлениях;

передачу информации из магистралей в регистры и обратно;

проверку и формирование контрольных кодов передаваемой информации.

Структурная схема К583ВА1 (см. рис. 1) содержит:

две типовые двунаправленные 4-разрядные магистрали данных L1 [0...3], L2 [0...3] с «открытым» коллектором;

мощную двунаправленную 4-разрядную магистраль данных с «открытым» эмиттером $\overline{L3}$ [0...3];

два буферных 4-разрядных регистра хранения информации R1, R2;

семь входов синхронизации $\overline{S1}$...$\overline{S7}$ для стробирования передачи информации;

2-разрядную двунаправленную магистраль проверки и формирования контрольных кодов передаваемой информации с «открытым» эмиттером $\overline{A}$, $\overline{K}$;

схему паритетного контроля;

4-разрядный блок усилителей формирователей.

Информация в регистр R1 (R2) может быть записана с шин данных L1 (L2) или $\overline{L3}$ с последующей выдачей в эти же шины. Информация по направлению L1→R1→$\overline{L3}$ (L2→R2→$\overline{L3}$) или $\overline{L3}$→R1→L1 ($\overline{L3}$→R2→→L2) передается с инверсией, а по направлению L1→R1→L1 (L2→R2→L2) — без инверсии.

Режим работы микросхемы задается синхросигналами $\overline{S1}$...$\overline{S7}$ алгоритмом (см. рис. 2).

Синхросигнал $\overline{S7}$ определяет, направлен ли поток информации из магистралей L1, L2 в $\overline{L3}$ или обратно. При $\overline{S7}=0$ данные передаются из магистралей L1 и L2 в магистраль $\overline{L3}$. При $\overline{S7}=1$ разрешена передача данных из магистрали $\overline{L3}$ в любую из магистралей L1 и L2. Синхросигнал $\overline{S1}$ стробирует занесение информации с магистрали L1 в регистр R1, информация поступает на вход R1 при условии $\overline{S7}=0$. Синхросигнал $\overline{S2}$ разрешает вывод информации из R1 на шину L1 при условии $\overline{S7}=1$.

Синхросигнал $\overline{S3}$ стробирует запись информации с магистрали L2 в регистр R2, информация поступает на вход R2 при условии $\overline{S7}=0$. Синхросигнал $\overline{S4}$ разрешает вывод информации из регистра R2 на магистраль L2 при $\overline{S7}=1$.

Синхросигнал $\overline{S5}$ адресует регистр R1 при записи с магистрали $\overline{L3}$ и при выводе на магистраль $\overline{L3}$.

Синхросигнал $\overline{S6}$ адресует регистр R2 при записи с магистрали $\overline{L3}$ и при выводе на магистраль $\overline{L3}$.

Магистраль содержит встроенную схему паритетного контроля информации магистрали $\overline{L3}$. При выводе информации на эту магистраль паритетная схема генерирует пятый контрольный разряд ($\overline{K}$) до «чет» или «нечет», а при вводе информации с этой магистрали паритетная схема анализирует на четность принимаемую информацию. При этом вывод $\overline{K}$ может дополнить контрольный разряд до «чет» или «нечет», и результат данных появится на выходе $\overline{A}$. Выводы $\overline{A}$ и $\overline{K}$ — двунаправленные, причем если они используются как выходы, то представляют собой эмиттерные повторители.

Микросхема позволяет организовать обмен информацией через линии связи длиной до 65 м (разрядность информационных посылок не ограничена).

Микросхема К583ВА2 (рис. 4) — магистральный приемопередатчик (МПП) без памяти — предназначена для согласования и обмена информацией между двумя типовыми одно-

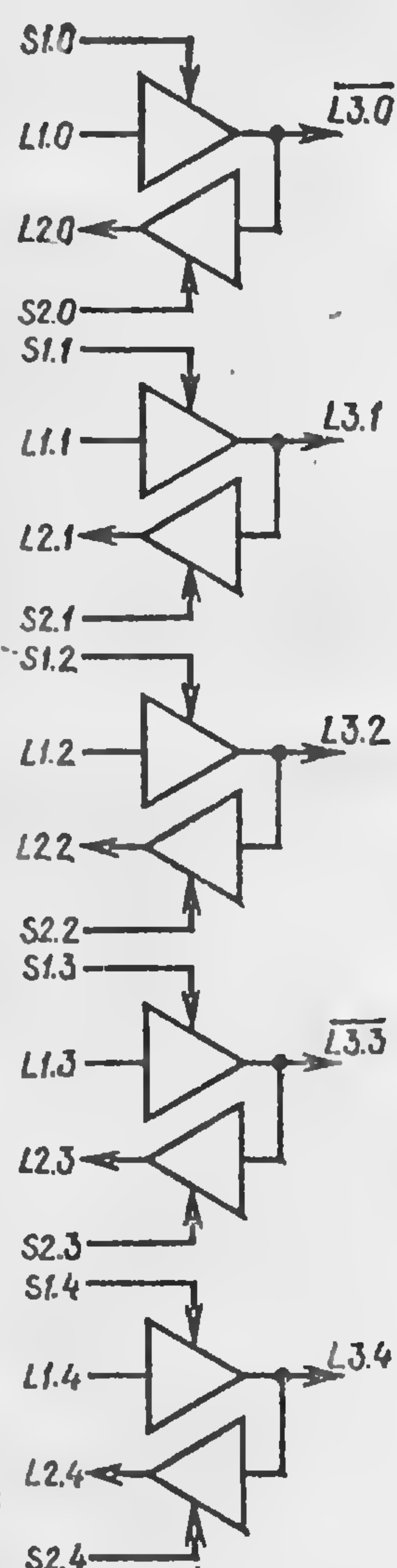


Рис. 4. Структурная схема микросхемы К583ВА2

направленными магистралями и мощной двунаправленной магистралью.

Микросхема обеспечивает (рис. 5) стандартные ТТЛ-уровни входных-выходных сигналов (см. ниже) для однонаправленных магистралей.

Конструктивно микросхема выполнена в 28-выводном металлокерамическом планарном корпусе.

МПП выполняет межмагистральный обмен данными между мощной магистралью и двумя типовыми однонаправленными магистралями в четырех направлениях.

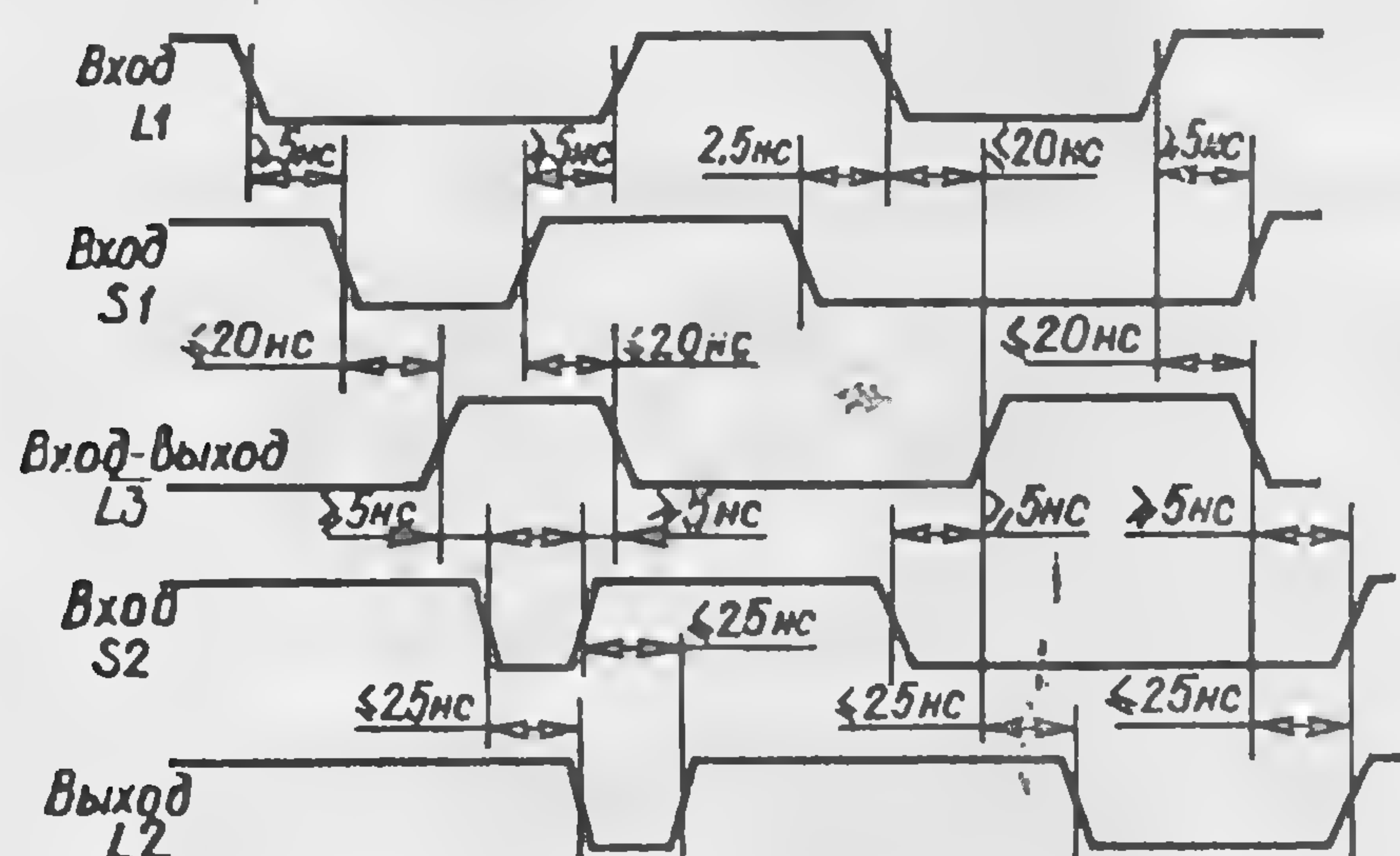


Рис. 5. Временная диаграмма работы микросхемы К583ВА2

Основные электрические параметры микросхемы К583ВА2

Напряжение питания U_{cc}, В	5±10 %
Ток потребления I_{cc}, мА, не более	80
Входной ток низкого уровня I_{IL}, мА, не более	|−0,2|
Входной ток высокого уровня I_{IH}, мкА, не более	50
Выходной ток низкого уровня I_{OL}, мА:	
для магистрали L2	20
для магистрали $\overline{L3}$	53
Типовое время задержки передачи информации между магистралями t_p, нс	10...25

Структурная схема МПП (см. рис. 4) содержит:

пять входных одноразрядных информационных магистралей L1.0...L1.4;

пять выходных одноразрядных магистралей с «открытым» коллектором L2.0...L2.4;

пять двунаправленных одноразрядных информационных магистралей $\overline{L3.0}$...$\overline{L3.4}$ с «открытым» эмиттером;

пять входов синхронизации S1.0...S1.4 для стробирования передачи информации по соответствующим разрядам от L1 к $\overline{L3}$;

пять входов синхронизации S2.0...S2.4 для стробирования передачи информации по соответствующим разрядам от $\overline{L3}$ к L2;

пять блоков усилителей-формирователей Y1...Y5.

Информация со входов L1.0...L1.4 при наличии разрешающих сигналов S1.0...S1.4 («Лог. 0») передается на выходы $\overline{L3.0}$...$\overline{L3.4}$. При этом информация на выходах $\overline{L3}$ инвертирована по отношению к магистрали L1.

Информация с входов-выходов $\overline{L3.0}$...$\overline{L3.4}$ при наличии разрешающих сигналов на соответствующих входах S2.0...S2.4 передается на выходы L2.0...L2.4. При этом информация на выходах L2 инвертирована по отношению к информации на входах-выходах $\overline{L3}$.

Информация с входов L1.0...L1.4 при наличии разрешающих сигналов на соответствующих входах обеих синхросерий S1, S2 может быть передана на выходы L2.0...L2.4. При этом информация на выходах L2 «прямая» по отношению ко входам L1.

Таблица истинности для i-го разряда МПП для сигналов положительной логики приведена в табл. 1.

Таблица 1

Таблица истинности микросхемы К583ВА2

Вход L1.i	0	0	1	1	Вход L3.i	0	0	1	1
Вход S1.i	1	0	1	0	Вход S2.i	1	0	1	0
Выход L3.i	0	1	0	0	Выход L2.i	1	1	1	0

Микросхема К583ВА3 (рис. 6) — быстродействующий универсальный коммутатор магистралей (табл. 2) — предназначена для построения коммутаторов и мультиплексоров данных, буферных устройств хранения и логической обработки данных, устройств восстановления информации в системах с резервированием.

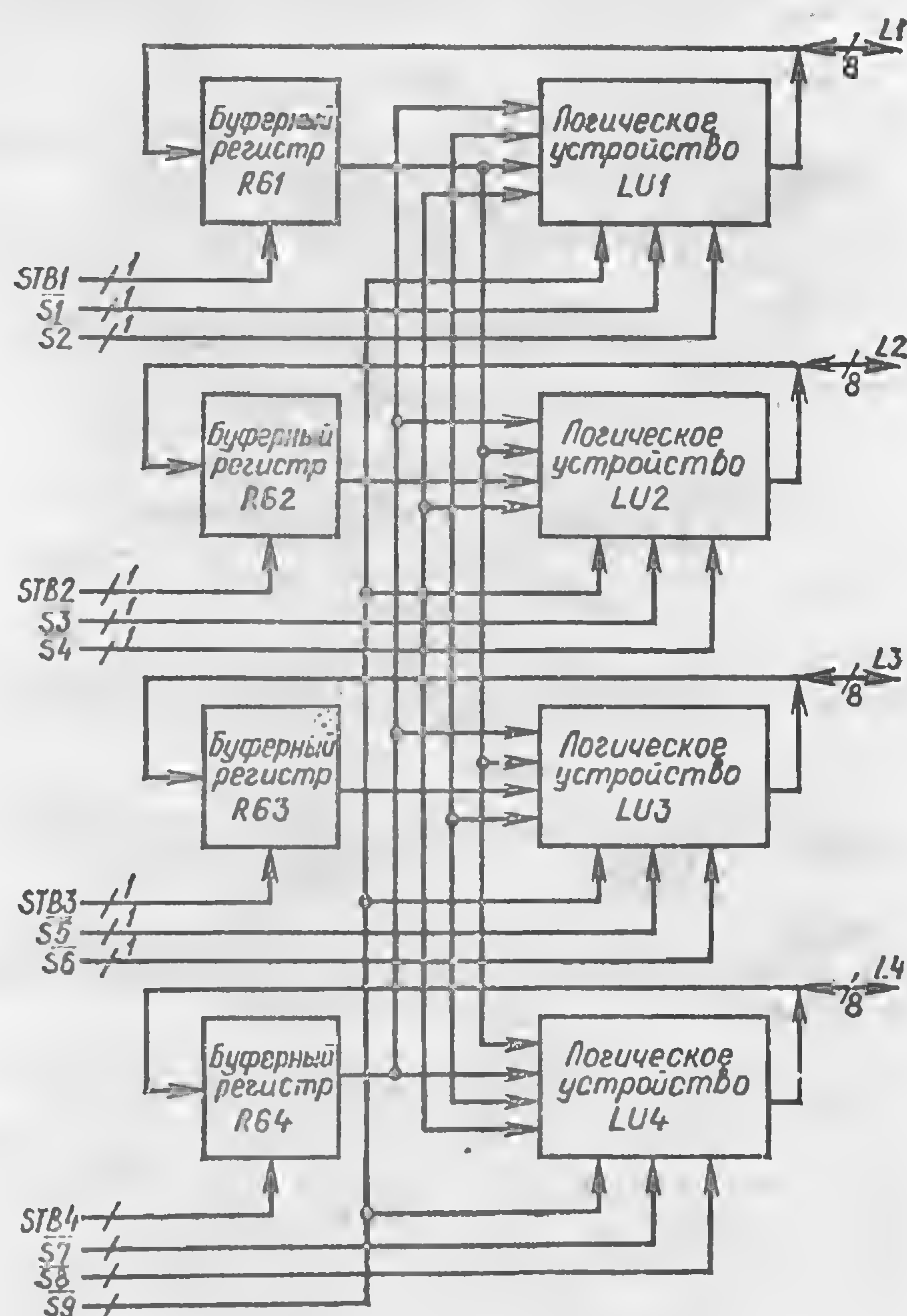


Рис. 6. Структурная схема микросхемы К583ВА3

Микросхема обеспечивает (рис. 7) стандартные ТТЛ-уровни входных-выходных сигналов (см. ниже).

Конструктивно микросхема выполнена в 48-выводном металлокерамическом планарном корпусе.

Универсальный коммутатор магистралей (УК) выполняет:

межмагистральный обмен данными между четырьмя информационными магистралями в 12 направлениях;

передачу информации из магистралей в регистры;

передачу информации из регистров в магистрали;

логическую обработку байтовых данных;

Таблица 2

Система микрокоманд микросхемы К583ВА3

Мнемоника микро-команды	Код микрокоманды			Выполняемая операция			
	S9	SI	SI+1	(I=1)	(I=3)	(I=5)	(I=7)
NOP	0	0	0	TC → L1	TC → L2	TC → L3	TC → L4
MV	0	0	1	R62(L2) → L1	R61(L1) → L2	R61(L1) → L3	R61(L1) → L4
MV	0	1	0	R63(L3) → L1	R63(L3) → L2	R62(L2) → L3	R62(L2) → L4
MV	0	1	1	R64(L4) → L1	R64(L4) → L2	R64(L4) → L3	R63(L3) → L4
NOP	1	0	0	TC → L1	TC → L2	TC → L3	TC → L4
MV	1	0	1	R61 → L1	R62 → L2	R63 → L3	R64 → L4
AND	1	1	0	$\wedge$(R62, R63) → → L1	$\wedge$(R63, R64) → L2	$\wedge$(R61, R64) → L3	$\wedge$(R61, R62) → L4
VC	1	1	1	$\wedge$(L2, L3) → L1 $\vee$[$\wedge$[L2, L3) $\wedge$(L2, L4), $\wedge$(L3, L4)] → L1	$\wedge$(L3, L4) → L2 $\vee$[$\wedge$(L1, L3), $\wedge$[L3, L4), $\wedge$(L1, L4)] → L2	$\wedge$(L1, L4) → L3 $\vee$[$\wedge$(L1, L2), $\wedge$(L2, L4), $\wedge$(L1, L4)] → L3	$\wedge$(L1, L2) → L4 $\vee$[$\wedge$(L1, L2), $\wedge$(L2, L3), $\wedge$(L1, L3)] → L4

Примечания: 1) при I=1 коды приведены для S1, S2, при I=3 — для S3, S4, при I=5 — для S5, S6, при I=7 — для S7, S8; 2) $\vee$ — операция конъюнкции, $\wedge$ — операция дизъюнкции; 3) TC — третье состояние выхода магистрали.

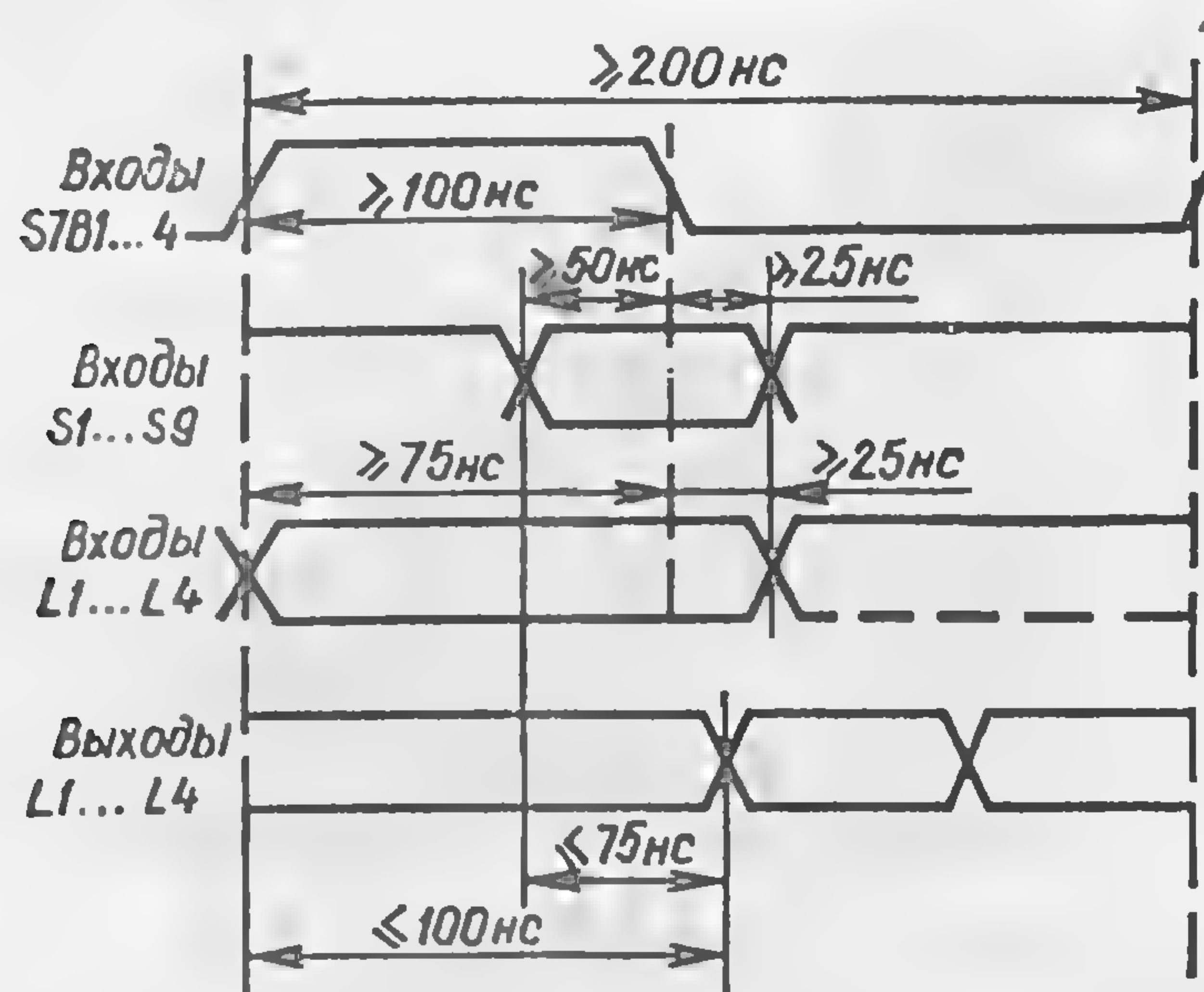


Рис. 7. Временна́я диаграмма работы микросхемы К583ВА3

одновременный или раздельный прием и одновременную или раздельную выдачу тактовой информации на 4 магистрали данных;

восстановление информации по мажоритарному принципу «2 из 3» из трех магистралей или регистров с выдачей результата на четвертую магистраль.

Основные электрические параметры микросхемы К583ВА3

Напряжение питания U_{cc}, В . . . . 5±10 %
Ток потребления I_{cc}, мА, не более 90
Выходное напряжение низкого уровня, U_{OL}, В, не более 0,5
Выходное напряжение высокого уровня U_{OH}, В, не менее 2,4
Входной ток низкого уровня I_{IL}, мА, не более:
для магистралей L1...L4 . . . . |−0,2|
для управляющих входов S1...S9 . . . |−0,4|
Входной ток высокого уровня I_{IH}, мкА, не более:
для магистралей L1...L4 50
для управляющих входов S1...S9 . . . 20
Выходной ток низкого уровня I_{OL}, мА . . . 4,0
Выходной ток высокого уровня I_{OH}, мА . . . |−0,4|
Время задержки передачи информации «магистраль — магистраль» t_p, нс, не более . . 100

Структурная схема К583ВА3 (см. рис. 6) содержит:
четыре 8-разрядные двунаправленные магистрали L1 [0...7]...L4 [0...7] с выходом с «тремя состояниями»;
четыре 8-разрядных буферных регистра R61...R64, синхронизируемых уровнем;
четыре 8-разрядных логических устройства LU1... ...LU4;
9-разрядную магистраль управления S1...S9;

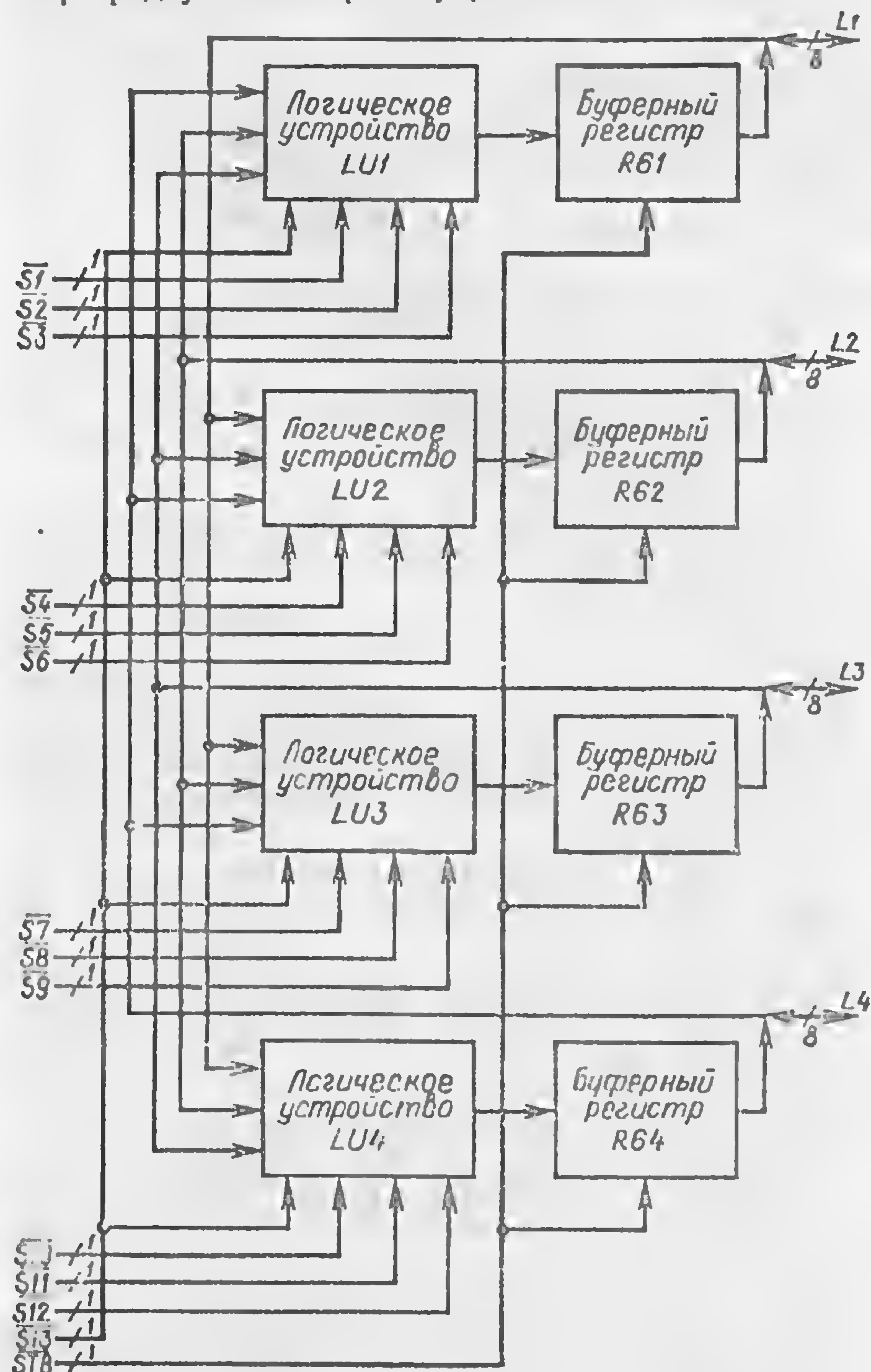


Рис. 8. Структурная схема микросхемы К583ВА4

Таблица 3

Система микрокоманд микросхемы К583ВА4

Мнемоники микрокоманды	Код микрокоманды S_I	S_{I+1}	S_{I+2}	S_{13}	Выполняемая операция (I=1)	(I=4)	(I=7)	(I=10)
NOP	0	0	0	0	00→L1	00→L2	00→L3	00→L4
NOP	0	0	1	0	00→L1	00→L2	00→L3	00→L4
NOP	0	1	0	0	00→L1	00→L2	00→L3	00→L4
NOP	1	0	0	0	00→L1	00→L2	00→L3	00→L4
NOP	0	0	0	1	00→L1	00→L2	00→L3	00→L4
MV	1	0	0	1	L2→L1	L1→L2	L1→L3	L1→L4
MV	0	1	0	1	L3→L1	L3→L2	L2→L3	L2→L4
MV	0	0	1	1	L4→L1	L4→L2	L4→L3	L3→L4
AND	1	1	0	0	∧(L2, L3)→L1	∧(L1, L3)→L2	∧(L1, L2)→L3	∧(L1, L2)→L4
AND	1	0	1	0	∧(L2, L4)→L1	∧(L1, L4)→L2	∧(L1, L4)→L3	∧(L1, L3)→L4
AND	0	1	1	0	∧(L3, L4)→L1	∧(L3, L4)→L2	∧(L2, L4)→L3	∧(L2, L3)→L4
OR	1	1	0	1	∧(L2, L3)→L1	∨(L1, L3)→L2	∧(L1, L2)→L3	∨(L1, L2)→L4
OR	1	0	1	1	∨(L2, L4)→L1	∨(L1, L4)→L2	∨(L1, L4)→L3	∨(L1, L3)→L4
OR	0	1	1	1	∨(L3, L4)→L1	∨(L3, L4)→L2	∨(L2, L4)→L3	∨(L2, L3)→L4
OR	1	1	1	1	∨(L2, L3, L4)→→L1	∨(L1, L3, L4)→L2	∨(L1, L2, L4)→L3	∨(L1, L2, L3)→L4
VC	1	1	1	0	∨[∧(L2, L3), ∧(L2, L4) ∧(L3, L4)]→L1	∨[∧(L1 L3), ∧(L1, L4), ∧(L3, L4)]→L2	∨[∧(L1, L2), ∧(L1, L4), ∧(L2, L4)]→L3	∨[∧(L1, L2), ∧(L1, L3), ∧(L2, L3)]→L4

Примечания: 1) управляющий сигнал R=0; 2) при I=1 коды приведены для S1...S3, при I=4 — для S4...S6, при I=7 — для S7...S9, при I=10 — для S10...S12; 3) ∧ — операция конъюнкции, ∨ — операция дизъюнкции; 4) для функционирования микросхемы согласно табл. 7 на магистрали, по которым поступает входная для схемы информация, должен выдаваться код 00_{16} из логических устройств, соответствующих этой магистрали.

4-разрядную магистраль синхронизации STB1...STB4.

Логические устройства синхронно выполняют логические операции над операндами, содержащимися в регистрах RG1...RG4 или поступающими непосредственно с магистралей L1...L4 (см. табл. 3). Результат операции логического устройства выдается на магистрали L1...L4. Микросхема выполняет операции мажоритирования байтовых данных для трех операндов.

Работа буферных регистров RG1...RG4 микросхемы синхронизирована синхросигналами STB1...STB4. Перепад уровней этих сигналов стробирует раздельное занесение информации в регистры данных RG1...RG4 соответственно.

Для микросхемы возможны два режима работы: при значении управляющих сигналов STB1...STB4=0 в регистрах RG1...RG4 состояния магистралей L1...L4 не сохраняются; при значении управляющих сигналов STB1...STB4=1 регистры RG1...RG4 сохраняют состояние магистралей L1...L4.

Микросхема К583ВА4 (рис. 8) — универсальный магистральный коммутатор (МК) байтовой информации (табл. 3) — предназначена для построения коммутаторов и мультиплексоров данных, буферных устройств хранения и логической обработки данных, устройств восстановления информации в системах с резервированием.

Микросхема обеспечивает (рис. 9) стандартные ТТЛ-уровни входных-выходных сигналов (см. ниже).

Конструктивно микросхема выполнена в 48-выводном металлокерамическом планарном корпусе.

Магистральный коммутатор выполняет:

межмагистральный обмен данными между четырьмя информационными магистралями в 12 направлениях;

передачу информации из магистралей в регистры;

передачу информации из регистров в магистрали;

логическую обработку байтовых данных;

одновременные прием и выдачу байтовой информации на четыре магистрали данных;

восстановление информации по мажоритарному принципу «2 из 3» из трех магистралей с выдачей результата на четвертую магистраль или записью в регистр четвертой магистрали.

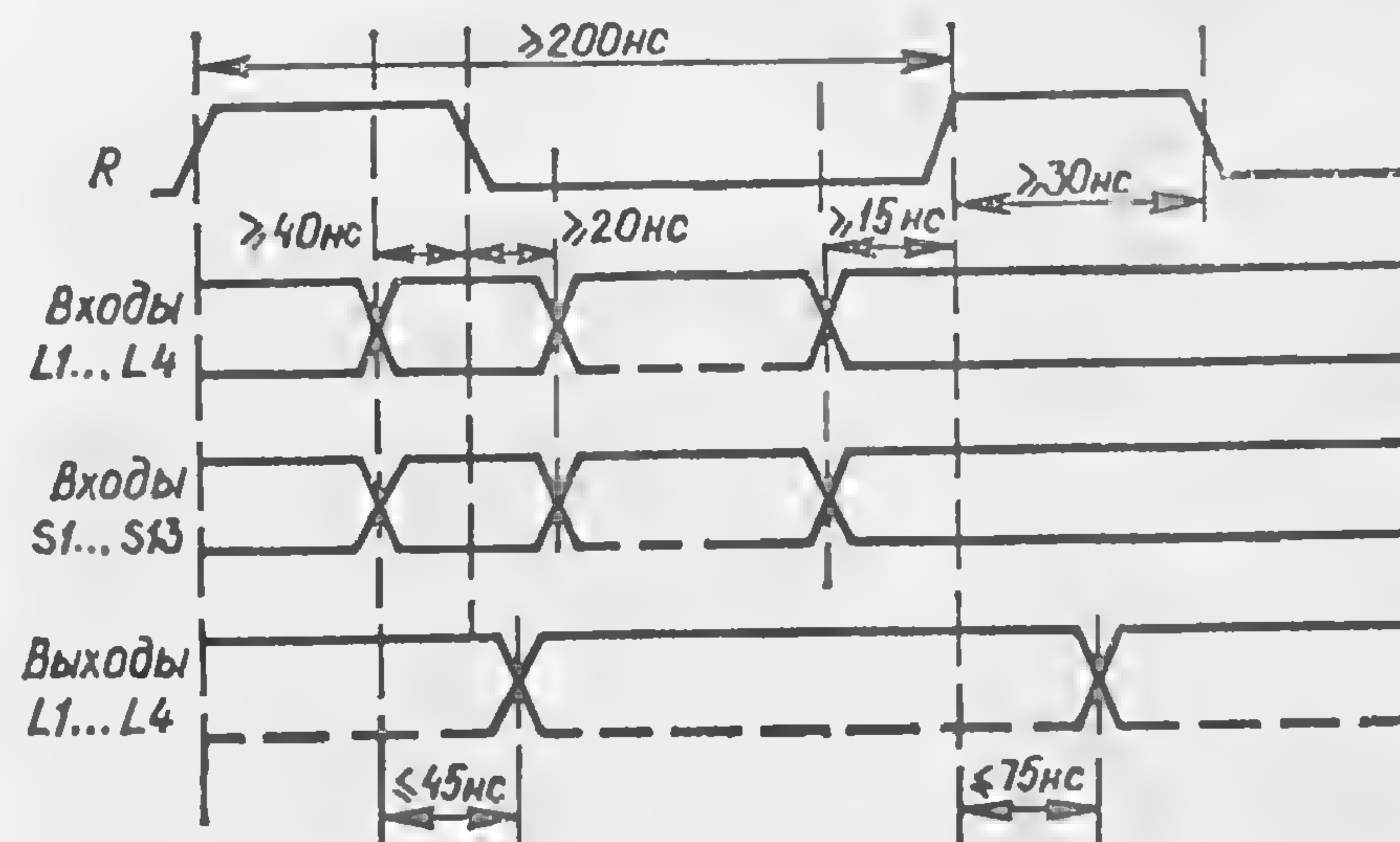


Рис. 9. Временная диаграмма работы микросхемы К583ВА4

Основные электрические параметры микросхемы К583ВА4

Напряжение питания U_{cc}, В . . . . 5В±10%

Ток потребления I_{cc}, мА, не более 120

Выходное напряжение низкого уровня U_{OH}, В, не более 0,5

Выходной ток высокого уровня I_{OH}, мА, не более 0,5

Входной ток низкого уровня I_{IL}, мА, не более:

для магистралей L1...L4 . . . . |−0,2|

для управляющих входов S1...S13, STB |−0,4|

Входной ток высокого уровня I_{IH}, мкА, не более:

для магистралей L1...L4 50

для управляющих входов S1...S13, STB 20

Выходной ток низкого уровня I_{OL}, мА . . . 20

Время задержки передачи информации «магистраль — магистраль» t_p, нс, не более . . 45

Продолжение см. на стр. 70.

От редакции. В этом номере журнала публикуются описание и технические характеристики четырех из семи микропроцессорных БИС для цифровой обработки сигналов: универсального процессорного элемента К1815ВФ1, сумматора последовательных чисел К1815ИМ1, накапливающего сумматора с интерфейсом К1815ВФ2, микропроцессора быстрого преобразования Фурье К1815ВФ3.

Остальные БИС комплекта серии К1815: быстродействующее АЛУ К1815ИД1, преобразователь кодов К1815ПР1 и микросхема ортогональной регистровой памяти К1815ИР1 будут рассмотрены в следующем номере.

УДК 681.325.5-181.48

Э. П. Калошкин, Л. В. Васильев

МИКРОПРОЦЕССОРНЫЙ КОМПЛЕКТ БИС ДЛЯ КОНВЕЙЕРНЫХ СИСТЕМ ЦИФРОВОЙ ОБРАБОТКИ СИГНАЛОВ

Совершенствование архитектуры и технологической базы современных БИС позволяет создавать системы цифровой обработки сигналов (ЦОС) в диапазоне от нуля до десятков мегагерц, работающие в реальном масштабе времени [1, 2]. В настоящее время достаточно четко определились такие области применения ЦОС, как обработка сигналов в радио-, гидро- и звуколокации, анализ спектра, сжатие данных в телеметрии, обнаружение полезных сигналов на фоне помех, фазовая автоподстройка частоты в дальней космической связи, адаптивная коррекция каналов связи, синтез и анализ речевых сигналов в телефонии, цифровые методы измерений, обработка сигналов в геологоразведке, метеорологии, сейсмологии и телевидении, в ТВ-адаптерах персональных ЭВМ и т. д.

Наиболее распространенными элементами систем ЦОС являются цифровые фильтры (ЦФ), которые можно рассматривать как самостоятельное направление цифровой техники [3]. Большой интерес разработчиков к ЦФ обусловлен рядом их преимуществ по сравнению с аналоговыми: отсутствие реактивных элементов в конструкции, высокая точность прогнозирования и реализации основных характеристик, возможность программной (микропрограммной) перестройки, изменения ширины полосы пропускания и характеристической частоты.

Конвейерная организация процесса обработки данных, широко используемая при решении задач ЦОС, позволяет совмещать во времени выполнение различных операций и обеспечивает высокую производительность.

Известные цифровые БИС, применяемые для построения систем ЦОС [4], можно разделить на следующие классы:

1. Микропроцессорные БИС общего назначения, к которым подключаются внешние АЦП и ЦАП. Для увеличения эффективности работы микропроцессоров общего назначения выпускаются так называемые «математические» кристаллы или вспомогательные арифметические процессоры.

2. Однокристальные микроЭВМ с встроенными АЦП и ЦАП (i8022, i8051, i8096, TMS2100 и др.), предназначенные для обработки сигналов в диапазоне до 10 МГц. Система команд БИС данного класса не предусматривает специальных функций ЦОС (умножение комплексных чисел, масштабирование и т. д.). Эти БИС позволяют реализовать простейшие функции сбора и частичной обработки аналоговых данных.

3. Специализированные комплекты микропроцессорных БИС (Am2920, S2811, μPD7720, F29500), в состав которых входят центральные процессорные элементы (цифровые процессоры сигналов), ориентированные на решение задач ЦОС. Микропрограммная организация БИС в этом случае является обязательной, поскольку исключает необходимость дополнительных затрат времени на дешифрацию команд — коды операций поступают непосредственно на операционные блоки БИС из ПЗУ микрокоманд. Использование микропрограммного управления наряду с конвейерными принципами обработки информации позволяет обеспечить достаточно высокую производительность систем. В этом классе БИС следует выделить семейство быстродействующих биполярных кристаллов F29500, предназначенных для реализации алгоритмов быстрого преобразования Фурье (БПФ), цифровой фильтрации, перемножения матриц, позволяющих решать задачи обработки сигналов на высоких частотах.

Существенным недостатком быстродействующих биполярных БИС ЦОС, реализуемых обычно на основе ЭСЛ-схем, является сравнительно большая потребляемая мощность (3...5 Вт/корпус). Один из эффективных путей ее уменьшения — использование маломощных схем на основе транзисторно-транзисторной логики с диодами Шотки (ТТЛШ) и интегральной инжекционной логики ($И^2Л$).

Микропроцессорные БИС серии К1815 представляют собой элементную базу для построения высокопроизводительных систем и устройств ЦОС с конвейерными принципами

организации информационных потоков. Особенностью БИС этой серии является возможность эффективной реализации на их основе алгоритмов вычисления дискретного (ДПФ) и быстрого (БПФ) преобразований Фурье, алгоритмов свертки и т. д.

В состав комплекта серии К1815 входят следующие БИС:

К1815ВФ1 — универсальный процессорный элемент;

К1815ИМ1 — быстродействующий 16-разрядный сумматор последовательных чисел;

К1815ВФ2 — накапливающий 24-разрядный сумматор с интерфейсом;

К1815ВФ3 — микропроцессор для построения процессора БПФ;

К1815ИД1 — быстродействующее арифметико-логическое устройство;

К1815ПР1 — преобразователь кодов последовательных чисел;

К1815ИР1 — микросхема ортогональной регистровой памяти.

Микросхемы комплекта выполнены на основе маломощной транзисторно-транзисторной логики с диодами Шотки (ТТЛШ) и интегральной инжекционной логики ($И^2Л$) по базовой планарно-эпитаксиальной технологии с использованием методов ионного легирования, тонких эпитаксиальных пленок, изопланарной изоляции и двухуровневой металлизации.

По электрическим параметрам разработанные микросхемы полностью совместимы с микросхемами серий К589, К1533, К133, К155, К583. Высокая нагрузочная способность по входу ТТЛШ БИС серии К1815 обеспечивается применением во входных каскадах транзисторов p-n-p типа, а высокая нагрузочная способность по выходу и высокая крутизна фронтов выходных сигналов — реализацией выходных каскадов по схеме с «активной» нагрузкой. Рабочий диапазон температур БИС —10 °С...+85 °С.

ЛИТЕРАТУРА

1. Зарубежная электронная техника, 1983, № 1, с. 58—94.
2. Electronic Design, 1984, v. 32, № 4, p. 52.
3. Капеллини В., Константинидис А. Дж., Эмилиани П. Цифровые фильтры и их применение. — М.: Энергоатомиздат, 1983.
4. Зарубежная электронная техника, 1985, № 3, с. 3—40.

Статья поступила 30 августа 1985 г.

УДК 681.325.5:621.3.049.77

Э. П. Калошкин, А. И. Белоус, В. В. Палиенко, О. В. Подрубный, В. М. Журба

БИС УНИВЕРСАЛЬНОГО ПРОЦЕССОРНОГО ЭЛЕМЕНТА К1815ВФ1

БИС универсального процессорного элемента (УПЭ) осуществляет прием с входных магистралей данных двоичных операндов в последовательно-параллельном прямом коде (одновременно по два разряда каждого операнда в одном такте), умножение принятых операндов и суммирование полученных произведений или, в зависимости от управляющего сигнала, только перемножение принятых операндов и выдает результат этих операций в последовательно-параллельном дополнительном коде на выходные магистрали данных.

БИС УПЭ (рис. 1) может работать в двух основных режимах:

1. Вычисление величин C1=A1·B1+A2·B2, где A1, B1, A2, B2 — 8-разрядные сомножители, C1 — 16-разрядный результат. Выполняется при высоком уровне сигнала на управляющем входе выбора режима MO1.

2. Вычисление величин C=A·B, где A и B — 16-разрядные сомножители, C — 32-разрядный результат. Выполняется при низком уровне сигнала на входе MO1.

Обозначение и функциональное назначение выводов БИС УПЭ представлены в табл. 1. При выполнении режимов 1 и 2 имеется возможность не принимать новые значения операндов A1, A2 (A) в рабочие регистры БИС, а проводить операции над ранее принятыми значениями операндов. Для устранения эффекта переполнения при работе БИС УПЭ в режиме 1 предусмотрено автоматическое масштабирование выходных результатов путем их сдвига на один разряд вправо. При этом младший разряд теряется, а освободившийся старший разряд заполняется знаком результата.

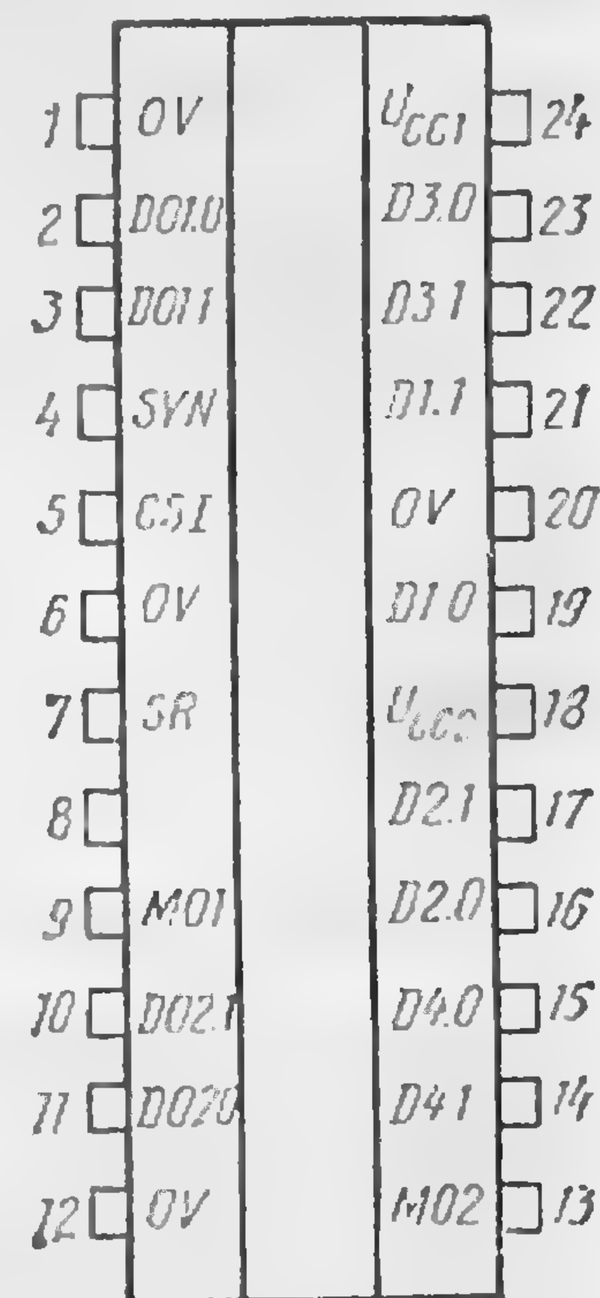


Рис. 1. Условное графическое изображение БИС К1815ВФ1

В состав БИС УПЭ входят (рис. 2):

четыре 2-разрядных входных магистрали данных D1...D4;

рабочие регистры PA1, PA2, PB1, PB2 для приема и хранения входных операндов;

блоки U1 и U2, обеспечивающие получение побитовых произведений входных операндов и управляющие записью операндов в рабочие регистры и выдачей побитовых произведений в блоки K1 и K2;

первый (K1) и второй (K2) каналы формирования результата из побитовых произведений;

схемы анализа знака UZ;

блок управления UC;

выходные 2-разрядные магистрали DO1 и DO2.

Входные магистрали данных D1...D4 предназначены для приема операндов в прямом коде (последовательно-

Таблица 1

Назначение выводов микросхемы К1815ВФ1

Вывод	Обозначение	Назначение
1	OV	Общий
2	D01.0	Выход младшего разряда результата из канала 1
3	D01.1	Выход старшего разряда результата из канала 1
4	SYN	Вход синхросигнала
5	SCI	Вход сигнала сопровождения знака
6	OV	Общий
7	SR	Сигнал общего сброса
8		Не используется
9	M01	Вход выбора режима работы
10	D02.1	Выход старшего разряда результата из канала 2
11	D02.0	Выход младшего разряда результата из канала 2
12	OV	Общий
13	M02	Вход выбора режима хранения
14	D4.1	Вход старшего разряда операнда B2
15	D4.0	Вход младшего разряда операнда B2
16	D2.0	Вход младшего разряда операнда A2
17	D2.1	Вход старшего разряда операнда A2
18	U_{cc2}	Напряжение питания +3 В
19	D1.0	Вход младшего разряда операнда A1
20	OV	Общий
21	D1.1	Вход старшего разряда операнда A1
22	D3.1	Вход старшего разряда операнда B1
23	D3.0	Вход младшего разряда операнда B1
24	U_{cc1}	Напряжение питания +5 В

параллельном). В режиме 1 восьмиразрядные операнды вводятся по магистралям D1...D4. В режиме 2 16-разрядные операнды вводятся по магистралям D1, D3 (с магистралей D2, D4 информация не принимается.) Операнды поступают на входные магистрали по два разряда в каждом такте, начиная с младших. Вслед за подачей информационных разрядов операндов по сигналу CSI на входы младших разрядов операндов поступают самые старшие информационные разряды, а на входы старших разрядов поступают знаки операндов.

Регистры PB1, PB2 являются сдвиговыми. В режиме 1 регистр PB1 принимает информацию с магистрали D3, регистр PB2 — с магистрали D4. Перестройка в режим 2 осуществляется объединением регистров PB1 и PB2 в один сдвиговый регистр двойной длины, который принимает информацию с магистрали D3.

Регистры PA1, PA2 — статические. Запись в них производится по мере поступления информации в последовательно-параллельном коде (по два разряда в каждом такте). В такте 1 информация заносится в первую пару разрядов PA1 (PA2) после подачи сигнала CSI, в такте 2 — во вторую пару и т. д. Управление записью осуществляется распределителем, входящим в состав блоков U1, U2. В режиме 1 регистр PA1 принимает информацию с магистрали D1, регистр PA2 — с магистрали D2. Перестройка в режим 2 осуществляется при объединении регистров PA1, PA2 в один общий регистр двойной длины, который принимает информацию с магистрали D1.

Запись в регистры PA1, PA2 происходит при низком уровне на входе сигнала выбора режима хранения MO2. При высоком уровне сигнала на этом входе информация с магистралей D1, D2 в регистры PA1, PA2 не принимается, а умножение вновь поступающих операндов B1, B2 (B) производится на ранее введенные в регистры PA1, PA2 операнды A1, A2 (A).

При выполнении операций над n-разрядными операндами в режимах 1 и 2 получается результат двойной длины. Так как операнды поступают непрерывно, то для выдачи результата двойной длины используются два канала обработки данных. Блоки U1 и U2 формируют побитовые произведения разрядов входных операндов и выдают их в канал K1 для нечетных (после подачи сигнала SR) и в канал K2 для четных пар операндов. Каналы K1 и K2 одинаковы и представляют собой каждый двоичное дерево 2-разрядных полных сумматоров с запоминанием переноса. Для выполнения операции суммирования в режиме 1 и выдачи результата в дополнительном коде в состав блоков K1 и K2 введены преобразователи прямого кода в дополнительный. Кроме того, каналы K1 и K2 содержат по одному блоку нормализации, которые в режиме 1 масштабируют результат с коэффициентом 2^{-1}. Результаты, сформированные каналами K1 и K2, выдаются на выходные магистрали D01 и D02 соответственно.

Блок знака UZ обрабатывает знаковые разряды входных операндов и управляет преобразователями кодов, входящими в блоки K1 и K2. Блок управления UC выдает внутренние сигналы синхронизации S1, S2, а также сигналы V, которые управляют работой блоков U1, U2, осуществляют коммутацию входных магистралей на рабочие регистры, управляют работой схемы знака UZ, преобразователей кодов и блоков нормализации.

Работа БИС УПЭ синхронизируется двумя внешними

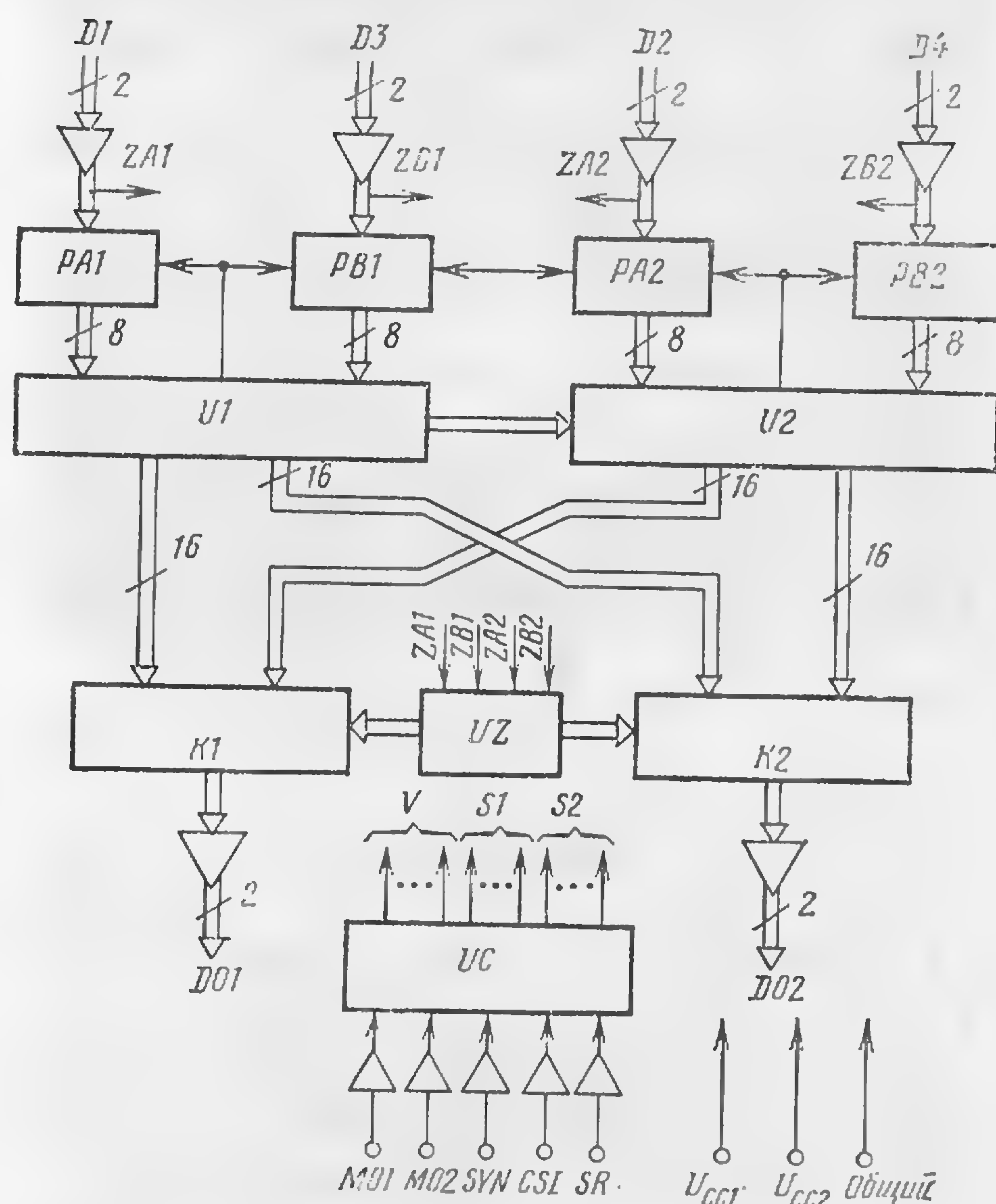


Рис. 2. Структурная схема БИС К1815ВФ1

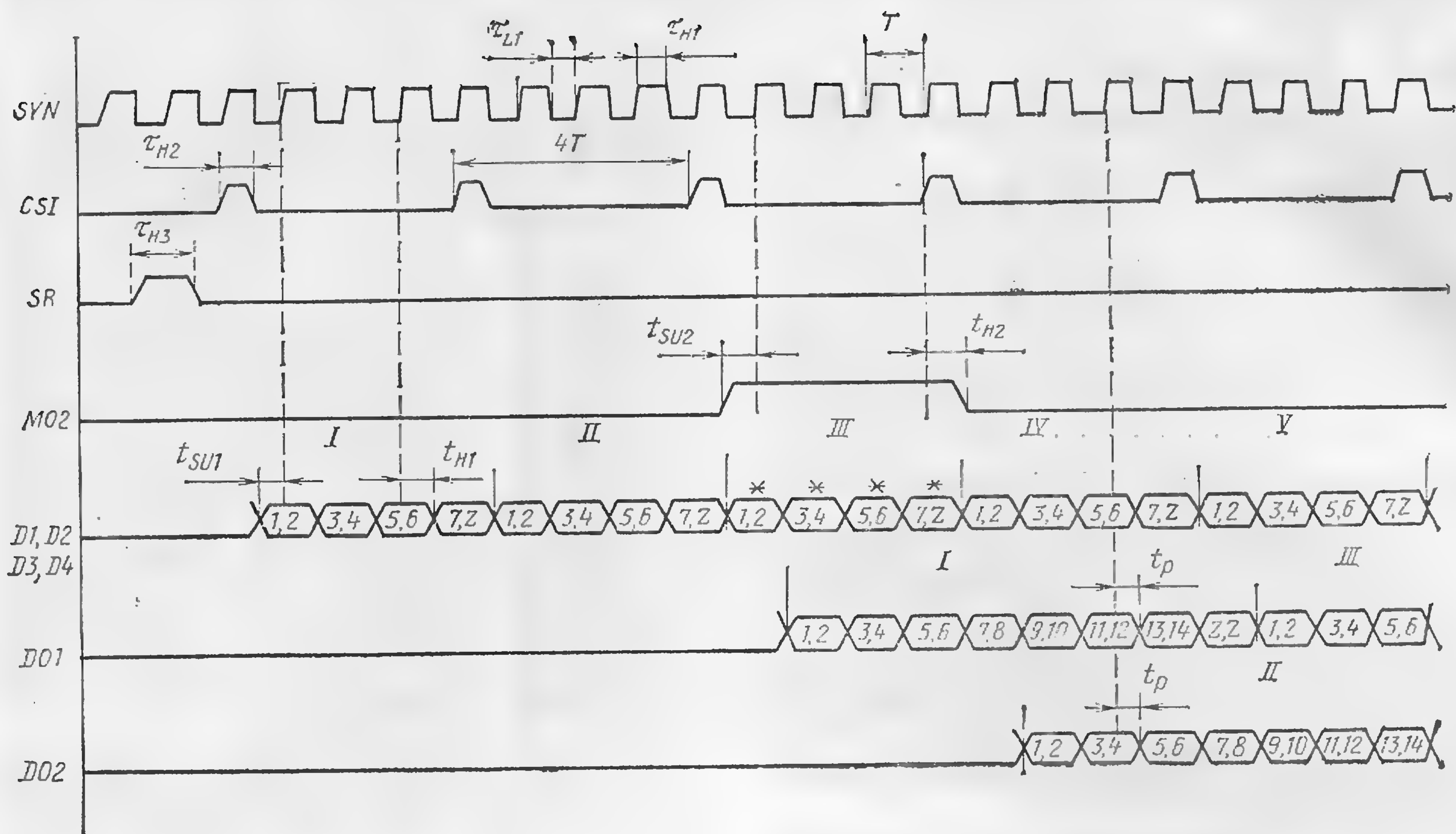


Рис. 3. Временная диаграмма работы БИС К1815ВФ1

синхросигналами SYN и CSI (рис. 3). Запись в регистры БИС УПЭ осуществляется положительными фронтами синхросигнала SYN. Синхросигналы подаются на входные магистрали одновременно со знаками операндов. Работа БИС начинается с подачи сигнала SR

Таблица 2

Динамические параметры БИС К1815ВФ1

Наименование	Обозначение	Норма, не менее
Период синхросигнала SYN	T	118
Длительность нижней полки синхросигнала SYN	τ_{L1}	70
Длительность верхней полки синхросигнала SYN	τ_{H1}	30
Длительность верхней полки сигнала CSI	τ_{H2}	15
Длительность верхней полки сигнала SR	τ_{H3}	10
Время предустанова информации на входах D1 ... D4 относительно переднего фронта синхросигнала SYN	t_{SU1}	30
Время удержания информации на входах D1 ... D4 относительно переднего фронта синхросигнала SYN	t_{H1}	50
Время предустанова сигнала на входе MO2 относительно переднего фронта синхросигнала SYN	t_{SU2}	5
Время удержания сигнала на входе MO2 относительно переднего фронта синхросигнала SYN	t_{H2}	40
Время задержки выдачи информации на выходные магистрали D01, D02 относительно переднего фронта синхросигнала SYN	t_P	100

установки в начальное состояние. После прихода первого импульса сигнала CSI необходимо вводить операнды для выполнения операции, заданной сигналами MO1, MO2. На рис. 3 римскими цифрами обозначены номера входных операндов и номера соответствующих им результатов на выходе БИС; * означает, что в отмеченном такте информация с магистралей D1, D2 в рабочие регистры БИС не принимается. Период следования импульсов синхросигнала SYNT⩾118 нс. Период синхросигнала CSI равен 4T в режиме 1 или 8T в режиме 2. Основные динамические параметры БИС представлены в табл. 2.

БИС работает в отрицательной логике, т. е. по информационным входам и выходам за «Лог. 0» принимается значение высокого уровня напряжения, за «Лог. 1» — значение низкого уровня.

БИС УПЭ выполнена по базовой планарно-эпитаксиальной технологии с использованием тонких эпитаксиальных пленок, окисной изоляции и двухуровневой металлизации. Основу БИС составляет четырехколлекторный инжекционный элемент ($И^2Л$), обеспечивающий реализацию функции ИЛИ—НЕ. Входные и выходные каскады выполнены по схемотехнике ТТЛ, что значительно улучшает динамические параметры и повышает нагрузочную способность БИС. Питание ТТЛ-каскадов осуществляется от источника напряжением U_{cc1} = =5 В±10 %. Для лучшего рассеяния мощности инжекционная часть микросхемы питается от отдельного источника напряжением U_{cc2}=3 В±5 %.

Основные электрические параметры БИС в диапазоне рабочих температур —10...+85 °С представлены ниже.

Электрические параметры БИС К1815ВФ1

Напряжение источника питания U_{cc1}, В . . . 5±10 %
Ток потребления I_{cc1}, мА, не более 60
Напряжение источника питания U_{cc2}, В . . . 3±10 %
Ток потребления I_{cc2}, мА, не более 300
Входное пороговое напряжение высокого уровня U_{IH}, В, не менее 2,0

Выходное напряжение высокого уровня U_{OH}, В, не менее 2,4
Входное пороговое напряжение низкого уровня U_{IL}, В, не более 0,8
Выходное пороговое напряжение U_{OL} низкого уровня I_{OL} при токе I_{OL}=8 мА, В, не более 0,5
Входной ток низкого уровня I_{IL}, мА, не более
по входу SYN . . . . |—0,8|
по остальным входам . . . . |—0,2|
Входной ток высокого уровня I_{IH}, мА, не более 40

По электрическим параметрам БИС полностью совместима с микросхемами серий ТТЛ и ТТЛШ.

Конструктивно БИС выполнена в металлокерамическом 24-выводном корпусе типа 4131.24-3 с планарным расположением выводов. По функциональным возможностям и конструктивному оформлению БИС К1815ВФ1 предназначена для применения в системах ЦОС, обладающих высокой производительностью, малыми габаритами и потребляемой мощностью. В частности, БИС К1815ВФ1 позволяет эффективно реализовать алгоритмы вычисления дискретного преобразования Фурье (ДПФ) и свертки, являющиеся основными алгоритмами ЦОС.

Статья поступила 30 августа 1985 г.

УДК 621.3.049.770:681.325.3

А. И. Милованов, А. В. Силин, А. М. Сахаров, Ю. В. Яковлев

БЫСТРОДЕЙСТВУЮЩИЙ СУММАТОР ПОСЛЕДОВАТЕЛЬНЫХ ЧИСЕЛ К1815ИМ1

Микросхема К1815ИМ1 представляет собой сумматор последовательных чисел (СПЧ) конвейерного типа, позволяющий выполнять операцию вычисления суммы последовательно-параллельных чисел. Сумматор выполнен на основе маломощной транзисторно-транзисторной логики с диодами Шотки (ТТЛШ) и содержит около 3000 элементов на кристалле площадью около 9,0 мм². Основные электрические параметры микросхемы в диапазоне температур —10...+85 °С приведены ниже.

Напряжение источника питания U_{cc}, В . . . 5 В±10 %
Ток потребления I_{cc}, не более 180
Входное напряжение, В:
низкого уровня, U_{IL}, не более 0,8
высокого уровня, U_{IH}, не менее 2
Входной ток, не более:
низкого уровня, I_{IL}, мА . . . . |—0,2|
высокого уровня, I_{IH}, мкА 40
Выходное напряжение, В:
низкого уровня, U_{OL}, не более 0,5
высокого уровня, U_{OH}, В, не менее . . . 2,4
Выходной ток, мА, не более:
высокого уровня, I_{OH} . . . . —0,4
низкого уровня, I_{OL} 4
Входной пробивной ток I_{IA}, мА, не более. . . 0,1

Конструктивно сумматор выполнен в стандартном 24-выводном плоском металлокерамическом корпусе. Условное графическое изображение БИС К1815ИМ1 приведено на рис. 1. Назначение выводов иллюстрируется таблицей. Микросхема обеспечивает суммирование восьми двоичных операндов неограниченной разрядности со знаком, представленных в последовательно - параллельном дополнительном коде (разряды подаются последовательно парами, младшими вперед, последний разряд знаковый). Входные и выходные данные представляются в отрицательной логике. Выдача результата, представленного в последовательно - параллельном модифицированном дополнительном коде, производится на выход-

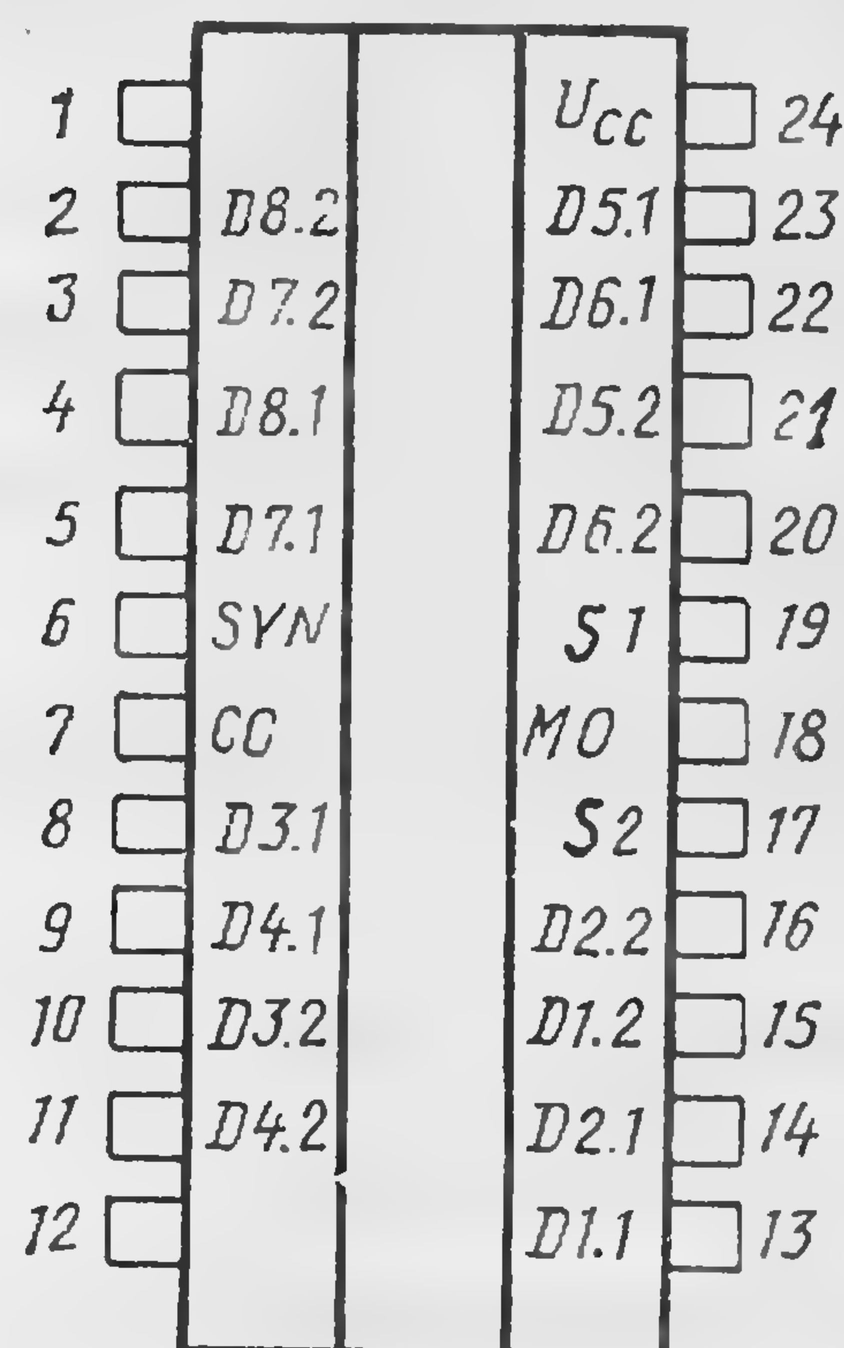


Рис. 1. Условное графическое изображение БИС К1815ИМ1

Назначение выводов микросхемы К1815ИМ1

Вывод	Назначение
1	Общий
2 ... 5	Входы магистрали данных: D8.2, D7.2, D8.1, D7.1
6	Вход сигнала синхронизации SYN
7	Вход признака сопровождения знака С0
8 ... 11	Входы магистрали данных: D3.1, D4.1, D3.2, D4.2
12	Общий
13 ... 16	Входы магистрали данных: D1.1, D2.1, D1.2, D2.2
17	Выход магистрали данных S2
18	Выход выбора режима работы МО
19	Выход магистрали данных S1
20 ... 23	Входы магистрали данных: D6.2, D5.2, D6.1, D5.1
24	Напряжение питания U_{cc}

ную 2-разрядную магистраль. Предусмотрена возможность выполнения нормализации результата (округление на два или три младших разряда) и размножения знака числа на два старших разряда.

Микросхема имеет восемь входных 2-разрядных магистралей данных D1...D8, выходную 2-разрядную магистраль результата DO, вход признака сопровождения знака СО2, вход управления режимом работы СО1, вход синхросигнала SYN, четыре последовательных сумматора первой ступени SM1.1...SM1.4, два последовательных сумматора второй ступени SM2.1, SM2.2, последовательный сумматор третьей ступени SM3.1, мультиплексор MUX и блок управления СО (рис. 2). Высокий уровень сигнала подается на вход СО2 в момент прихода знака. Подача высокого уровня на вход СО1 означает режим работы с размножением знака результата на два старших разряда. В связи с необходимостью

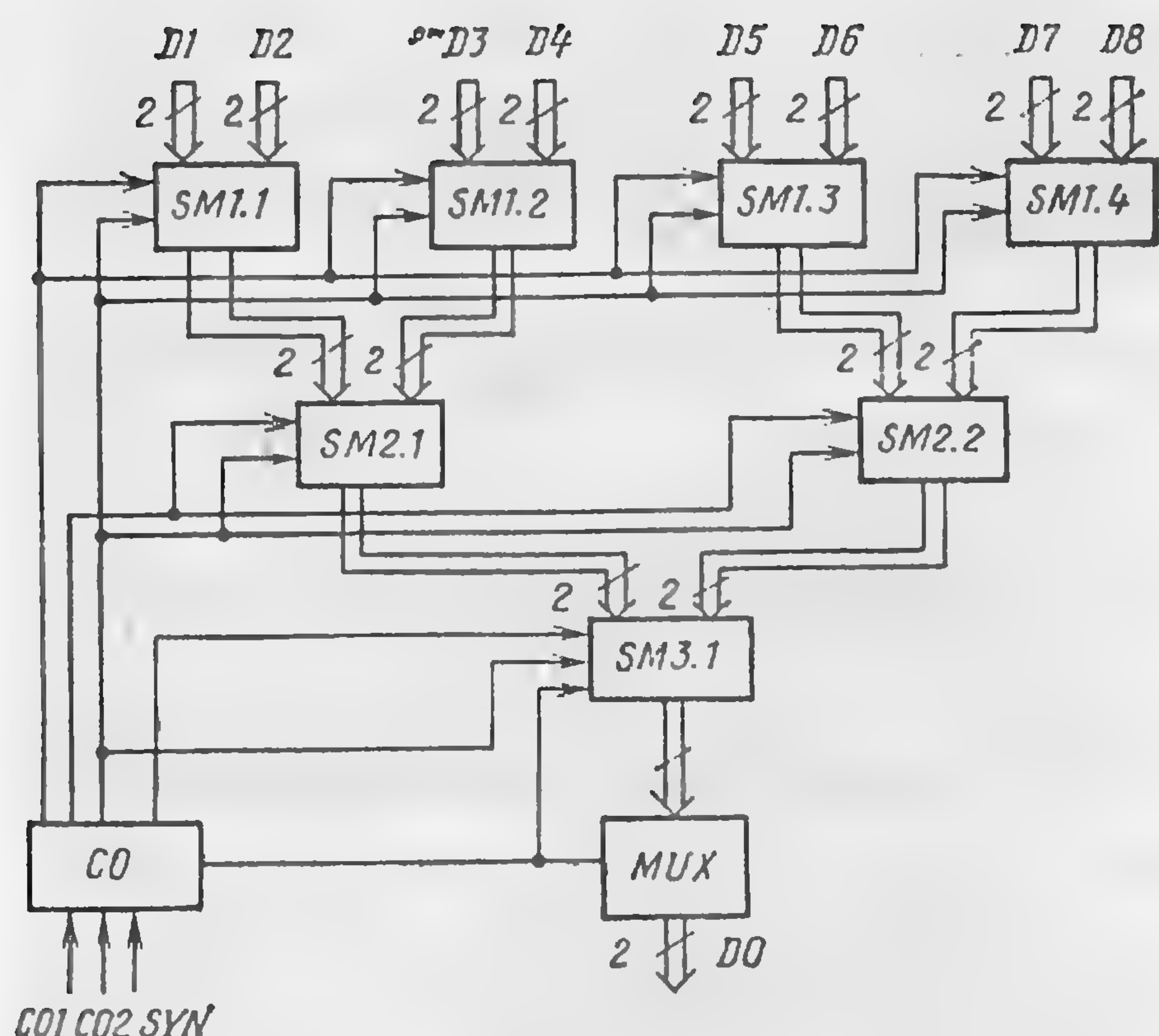


Рис. 2. Структурная схема сумматора

обеспечения высокого темпа выдачи результата на выходную магистраль (120 нс) в схему введены промежуточные триггеры, а вычисление суммы и соответственно структурная схема БИС реализованы в три ступени (по конвейерному принципу). Сумматоры всех ступеней имеют сходную структуру и содержат входные триггеры, полный сумматор младших разрядов, полный сумматор старших разрядов, триггер запоминания переноса, триггер задержки суммы старших разрядов, выходной мультиплексор. Сумматоры первой ступени SM1.1...SM1.4 выполняют суммирование соответствующих пар чисел, поступающих на входы D1...D8 (каждый сумматор суммирует два числа). Информация из сумматоров первой ступени поступает на входы сумматоров второй ступени SM2.1...SM2.2, которые вычисляют суммы соответственно первых двух и вторых двух пар чисел. Полученные на второй ступени суммы первых четырех и вторых четырех чисел поступают на сумматор третьей ступени SM3.1, который вычисляет конечный результат, поступающий через мультиплексор MUX на выходную 2-разрядную магистраль DO (на выход DO1 выдаются нечетные, на DO2 — четные разряды). Если CO1 = 1, результат выдается округленным на три младших разряда с двумя знаками в крайних старших разрядах. При CO1 = 0 результат суммирования выдается округленным на два младших разряда с одним знаком в старшем разряде.

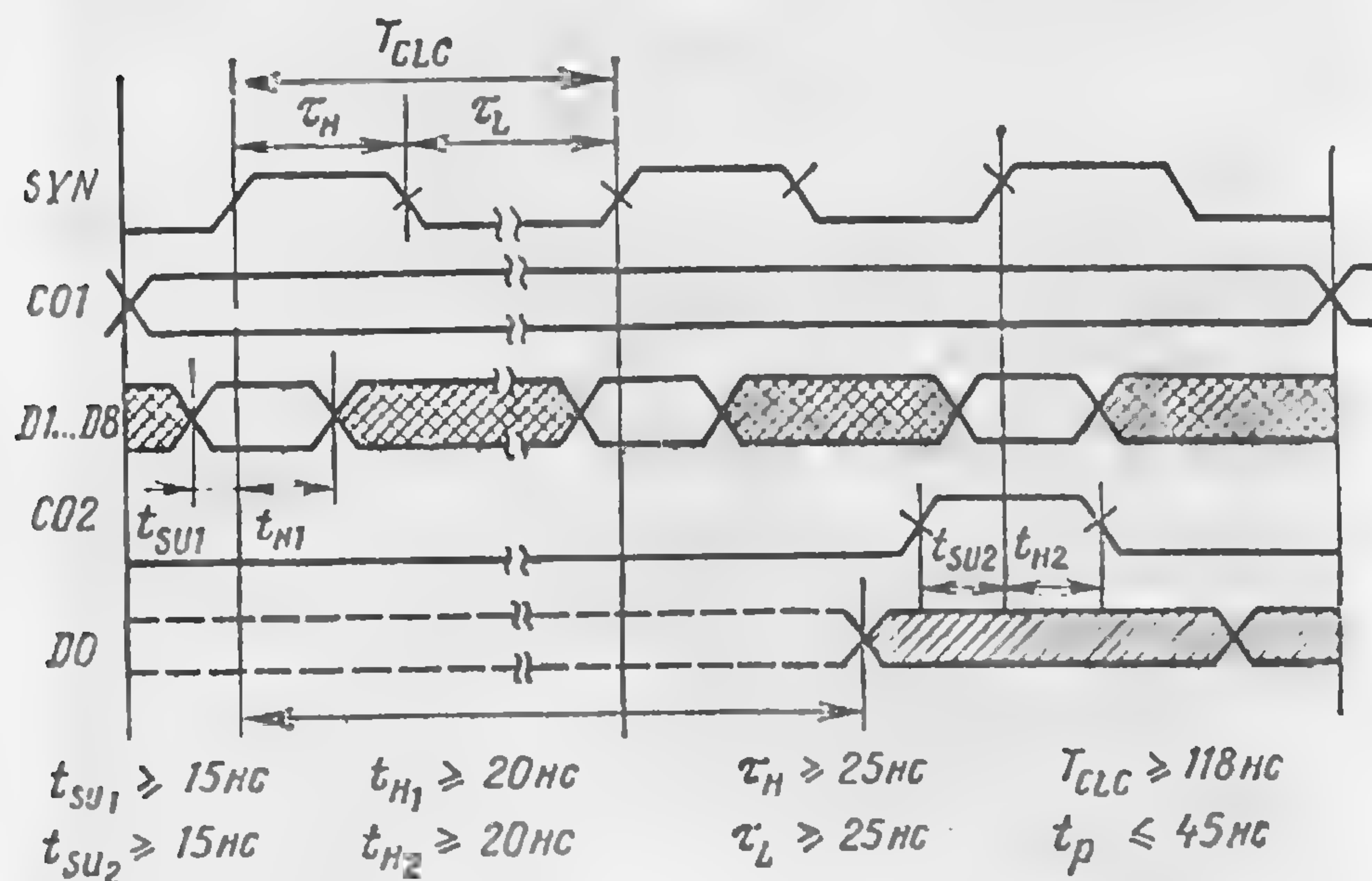


Рис. 3. Временная диаграмма работы сумматора.

Работа микросхемы синхронизируется внешним управляющим синхросигналом с периодом следования тактовых импульсов 120 нс. Упрощенная временная диаграмма работы сумматора приведена на рис. 3.

Базовым элементом микросхемы К1815ИМ1 является ТТЛШ элемент, выполняющий функцию И-ИЛИ-НЕ и обеспечивающий быстродействие не хуже 4 нс при рассеиваемой мощности не более 1 мВт. Базовым элементом памяти является R-S-триггер, тактируемый фронтом синхросигнала и выполненный по схеме M-S. Использование такой элементной базы позволило при типовой мощности рассеивания 500 мВт получить время задержки информации на выходную магистраль 20...25 нс.

БИС сумматора полностью совместима по электрическим параметрам с микросхемами К589, К133, К583, К584, К155, К1533 и обладает высокой нагрузочной способностью по входу, что обеспечивается применением во входных каскадах p-n-p транзисторов. Высокая крутизна фронтов сигналов, нагрузочная способность микросхемы по выходу достигаются выполнением выходов по схеме с «активной» нагрузкой.

Статья поступила 30 августа 1985 г.

УДК 681.325.5-181.48

А. И. Белоус, О. С. Вайнилович, В. М. Журба, Н. Ю. Сорокина, В. В. Шалимо

БИС НАКАПЛИВАЮЩЕГО 24-РАЗРЯДНОГО СУММАТОРА С ИНТЕРФЕЙСОМ

БИС накапливающего сумматора с интерфейсом К1815ВФ2 предназначена для построения широкого класса цифровых устройств и в первую очередь высокопроизводительных систем цифровой обработки сигналов (ЦОС). БИС К1815ВФ2 выполнена в 28-выводном корпусе. Условное графическое изображение микросхемы и назначение выводов показаны на рис. 1 и в табл. 1. К1815ВФ2 имеет 8-разрядную входную магистраль данных, 12-разрядную выходную магистраль результата, 3-разрядную магистраль адреса, вход синхросигнала, регистры RG1...RG7, мультиплексор SU1, дешифратор SU2, 24-разрядный сумматор SM (рис. 2). Базовым элементом БИС К1815ВФ2 является четырехколлекторный инжекционный элемент ($И^2Л$), выполняющий функцию ИЛИ-НЕ.

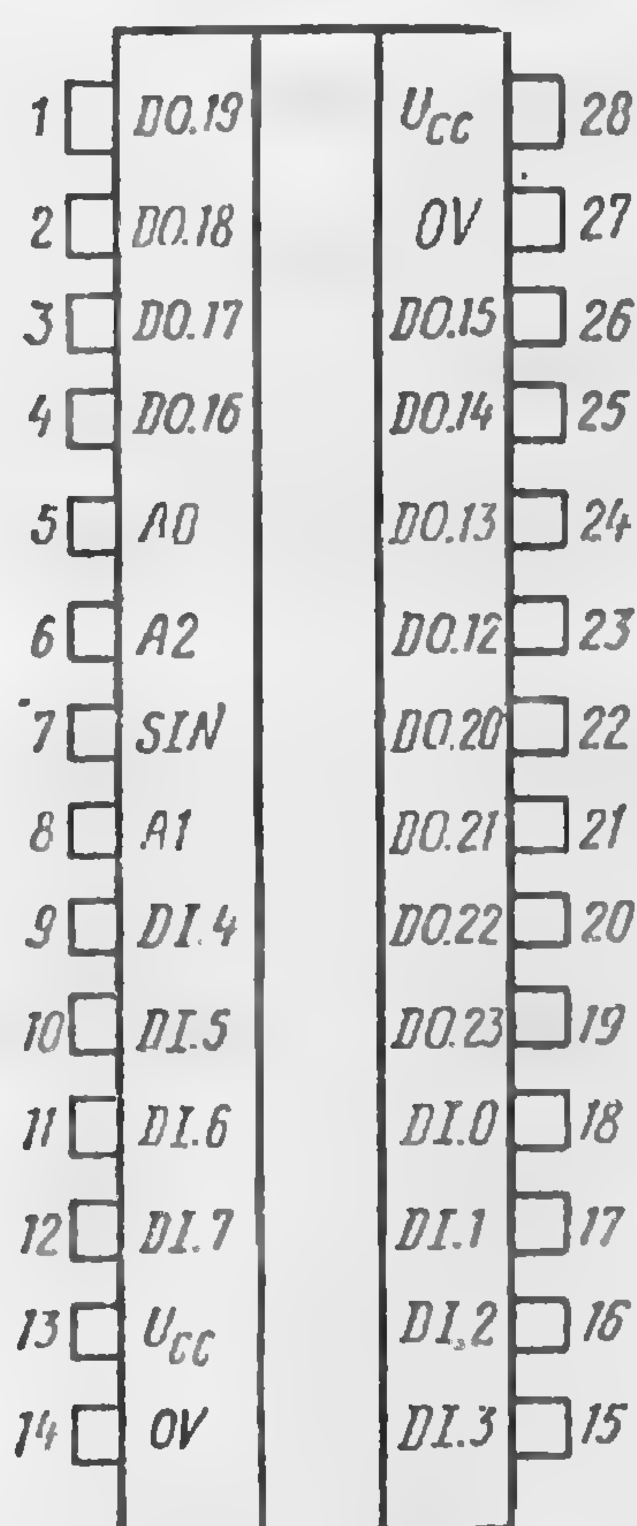


Рис. 1. Условное графическое изображение БИС К1815ВФ2

Управление работой БИС осуществляется 3-разрядной микрокомандой A0...A2, поступающей на дешифратор адреса D, который вырабатывает управляющие сигналы Y1...Y7, синхронизируемые синхронсигналом SIN. В табл. 2 приведены микрокоманды сумматоров.

Накапливающий сумматор SM осуществляет суммирование 24-разрядного аргумента с 16-разрядным приращением. Недостающие старшие разряды приращения представлены нулями. На выходную магистраль DO выдаются двенадцать старших разрядов результата суммирования.

Таблица 1

Назначение выводов БИС К1815ВФ2

Вывод	Назначение
1 ... 4	Выходы данных: D0.19 ... D0.16
5	Вход адреса A0
6	Вход адреса A2
7	Вход синхронизации SIN
8	Вход адреса A1
9 ... 12	Входы данных: DI.4 ... DI.7
13	Напряжение питания U_{cc}
14	Общий
15 ... 18	Входы данных: DI.3 ... DI.0
19 ... 26	Входы данных: D0.23 ... D0.15
27	Общий
28	Напряжение питания U_{cc}

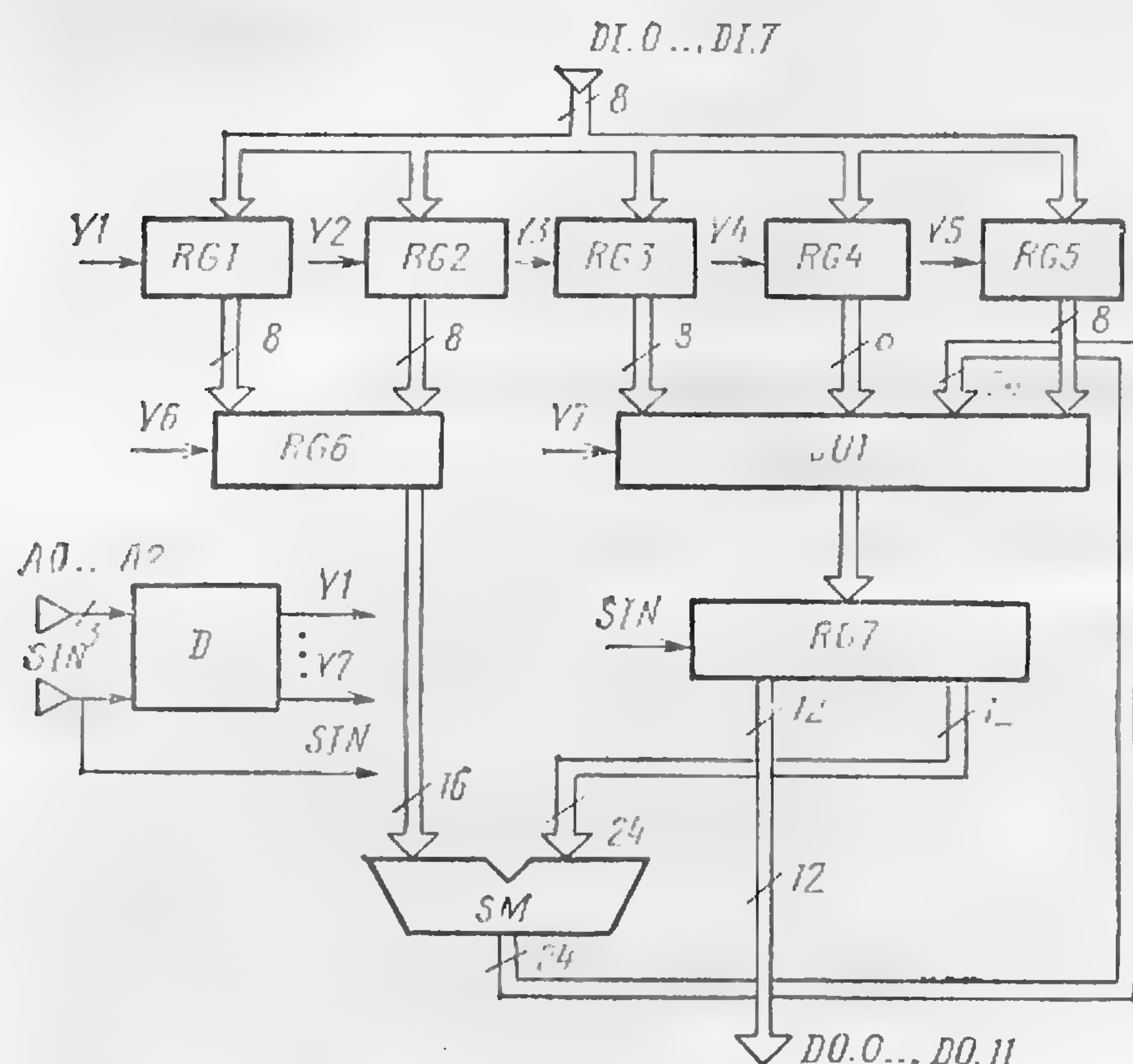


Рис. 2. Структурная схема БИС сумматора

Таблица 2

Микрокоманды БИС К1815ВФ2

Код микро-команды	Информация в регистре						
	RG1	RG2	RG3	RG4	RG5	RG6	RG7
Y1:000	DI	H	H	H	H	H	SM
Y2:001	H	DI	H	H	H	H	SM
Y3:010	H	H	DI	H	H	H	SM
Y4:011	H	H	H	DI	H	H	SM
Y5:100	H	H	H	H	DI	H	SM
Y6, Y7:101	H	H	H	H	H	RG1, RD2	RG3, RG1 RG5
X	H	H	H	H	H	H	SM

Примечание. Н — режим хранения информации предыдущего такта.

Одновременно со сложением предыдущих операндов в сумматоре SM и выдачей результата сложения на выходную магистраль DO можно записывать информацию во входные регистры RG для хранения и последующей выдачи информации на сумматор. Такая структура позволяет использовать БИС для построения цифровых систем с конвейерным принципом обработки данных. Временная диаграмма работы БИС в системе ЦОС представлена на рис. 3. Минимальный период следования синхроимпульсов $T_{CLC}=500$ нс.

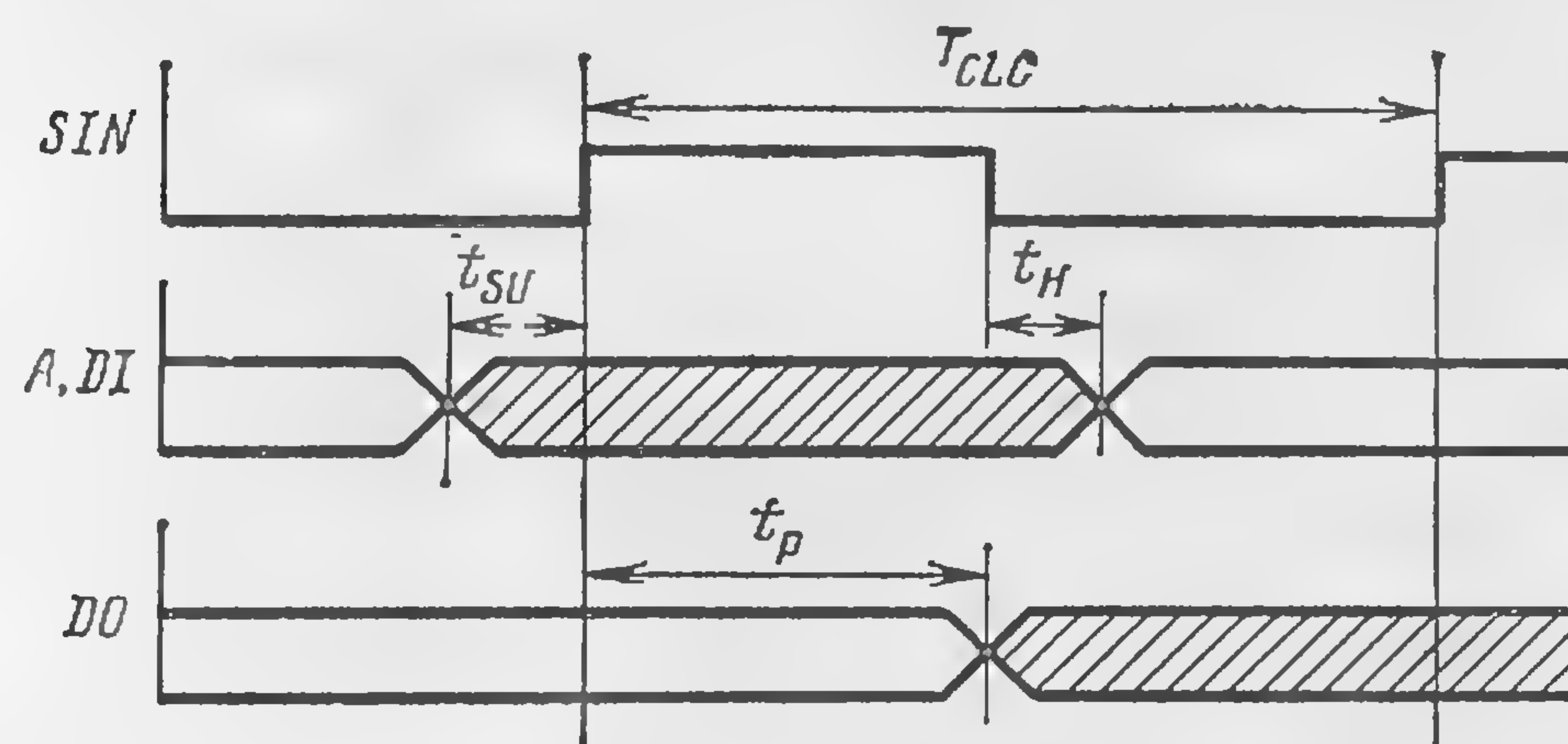


Рис. 3. Временная диаграмма работы БИС К1815ВФ2: T_{CLC} — период следования тактовых импульсов SIN; t_{SU} — время установки информации по входам DI.O...DI.7, A0...A2 относительно переднего фронта синхросигнала SIN; t_H — время удержания информации по входам DI.O...DI.7, A0...A2 относительно заднего фронта синхросигнала; t_p — время задержки выдачи информации на выходную магистраль DO.0...DO.11 относительно переднего фронта синхросигнала

Основные электрические параметры БИС в диапазоне рабочих температур —10...+85 °C представлены ниже. По электрическим параметрам БИС полностью совместима с микросхемами серий ТТЛ.

Электрические параметры БИС К1815ВФ2

Напряжение источника питания U_{cc}, В	+5±10%
Ток потребления I_{cc}, мА, не более	165
Входное напряжение, В:	
низкого уровня, U_{IL}, не более	0,8
высокого уровня, U_{IH}, не менее	2
Входной ток, не более	
низкого уровня, I_{IL}, мА	|—0,2|
высокого уровня, I_{IH}, мкА	40
Выходное напряжение, В	
низкого уровня, U_{OL}, не более	0,5
высокого уровня, U_{OH}, не менее	2,4
Выходной ток, мА, не более	
высокого уровня, I_{OH}	—0,4
низкого уровня, I_{OL}	8
Входной пробивной ток, I_{IA}, мА, не более	0,1
Период следования синхроимпульса T_{CLC}, нс, не более	472
Время задержки распространения t_p, нс, не более	350
Время установки сигнала информации по входам DI и A относительно среза сигнала SIN t_{SY}, нс, не более	100
Время сохранения сигнала информации по входам DI и A относительно фронта синхросигнала SIN t_H, нс, не более	60
Длительность импульса по входу синхронизации SIN τ_H, нс, не более	100

Применение схемотехники и технологии И²Л позволило реализовать схему с высокой плотностью упаковки. С целью повышения быстродействия и нагрузочной способности обрамление БИС (входные и выходные каскады) выполнено по схемотехнике ТТЛ. К достоинствам схемы следует отнести использование одного источника $U_{cc}=5$ В для питания инжекционных и ТТЛ блоков БИС.

Статья поступила 30 августа 1985 г.

УДК 621.372.542:681.31.002:621.382.8

В. А. Бобков, А. И. Сухопаров,
В. Ф. Гузик, А. В. Каляев, Ю. П. Попов,
О. Б. Станишевский

МИКРОПРОЦЕССОР К1815ВФ3 ДЛЯ ЦИФРОВОЙ ОБРАБОТКИ СИГНАЛОВ

Однокристальный микропроцессор К1815ВФ3 предназначен для построения устройств быстрого преобразования Фурье (БПФ), цифровых фильтров и систем цифровой обработки сигналов на их основе. Машинный язык микропроцессора является языком высокого уровня, а набор команд (см. таблицу) обеспечивает выполнение таких крупных операций, как БПФ, умножение и сложение комплексных чисел, нерекурсивный и рекурсивный фильтры (НРФ и РФ), полином, а также логических операций — модуль суммы или разности двух чисел, выделение большего числа из пары операндов. Для синхронизации данных, передаваемых между микропроцессорами в многопроцессорной системе, предусмотрена возможность программного управления временем задержки выдачи результата. При работе с числами, дающими переполнения, МП осуществляет автоматическое масштабирование. С этой целью все команды имеют дополнительный разряд Fj, кодирующий режим работы — с масштабированием или без него.

Код команды	Операция	
F5 F4 F3 F2 F1 F0	БАЛУ (S)	БУ (P)
X X X X X 0	2^{kd} (A * B)	2^{2d} (C+D) G
X X X X X 1	2^{kd} (A * B)	2^{2d} (C—D) G
0 X X X X X	$2^{k\delta}$ (A ± B)	$2^{2\cdot6}$ (C±D) G
1 X X X X X	$2^{k\delta}$ (A + B)	$2^{2\cdot5}$ (C+D) G
X 0 0 X X X	2^{1d} (A±B)	2^{2d} (C±D) G
X 0 1 X X X	2^{2d} (A±B)	2^{2d} (C±D) G
X 1 0 X X X	2^{3d} (A±B)	2^{2d} (C±D) G
X 1 1 0 0 X	$2^{1d}{}_{max}$ (A±B)	2^{2d} (C±D) G
X 1 1 0 1 X	$2^{2d}{}_{max}$ (A±B)	2^{2d} (C±D) G
X 1 1 1 0 X	$2^{3d}{}_{max}$ (A±B)	2^{2d} (C±D) G
X X X X 0 X	2^{kd} (A±B)	2^{2d} (C±D) G
X X X X 1 X	2^{kd} (A—B)	2^{2d} (C±D) G
X X X 0 X X	2^{kd} (A±B)	2^{2d} (C±D) G
X X X 1 X X	2^{kd} (A±B)	2^{2d} (C±D) G

Примечание. * ∈ {+, —, |...|, max}; |...| ∈ {|...|, (...)}; d=5; 6 — задержка результата в тактах; k=1, 2, 3 — коэффициент задержки.

БИС К1815ВФ3 выполнена по изопланарной технологии инжекционных логических схем, обеспечивающих высокую степень интеграции. Конструктивно оформлена в 24-выводной корпус. БИС МП полностью согласуется по электрическим характеристикам с ТТЛ-микросхемами. Выходные каскады выполнены в виде стандартных схем с открытым коллектором. Ниже приведены электрические параметры МП К1815ВФ3.

Основные электрические параметры

Ток питания, мА	300
Тактовая частота, МГц	10
Напряжение питания, В	1,5...5
Рассеиваемая мощность, мВт	360
Выходной ток «Лог. 0», мА	10
Время цикла, нс	100

Синхронизация работы МП осуществляется одной синхросерией тактовых импульсов. Производительность МП — 1...2 млн. арифметических или 100 тысяч крупных операций над 24-разрядными операндами.

В основу построения микропроцессора положены идеи реализации крупных операций методами программирования структуры [1], а также способ обработки цифровой информации старшими разрядами вперед. МП состоит из блока арифметическо-логических операций (БАЛО), блока умножения (БУ) и блока управления операциями (БУО) (рис. 1). Имеет пять информационных входов (LA, LB, LC, LD, LG) и два информационных выхода (LS, LP), что обеспечивает одновременное выполнение трех арифметическо-логических операций над пятью операндами (A, B, C, D, G) с одновременной выдачей двух результатов (S и P). Совокупность этих операций представляет собой унифицированную часть крупных операций. Операнды и результаты в МП могут иметь произвольную разрядность, точность умножения МП составляет 26 двоичных разрядов.

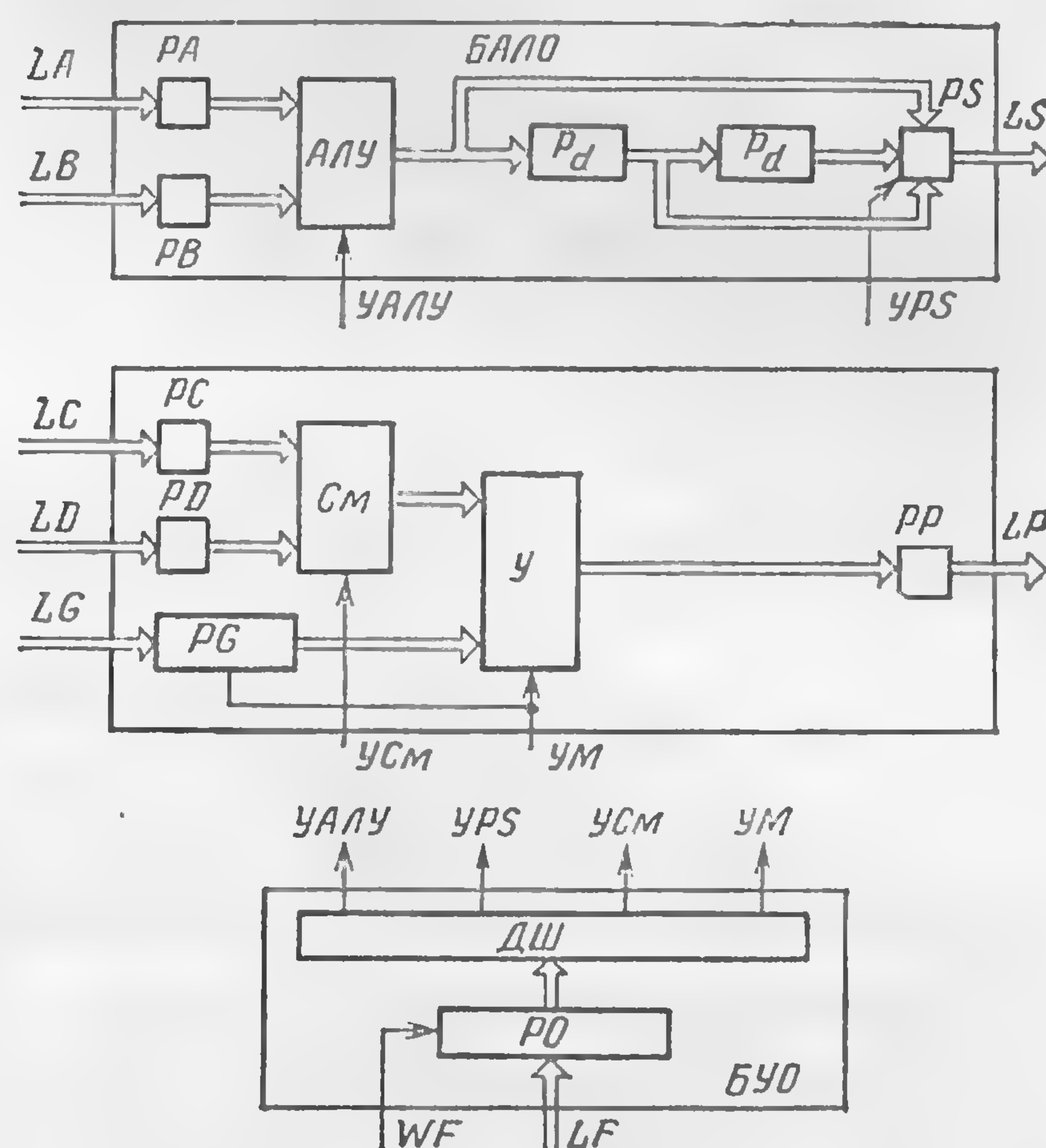


Рис. 1. Структурная схема микропроцессора К1815ВФ3

Блок арифметическо-логических операций в зависимости от заданного кода команды F осуществляет сложение или вычитание операндов A и B и выдает на выход либо их сумму (разность) с взятием (без взятия) модуля, либо бóльший из операндов. Работает БАЛО по конвейерному принципу. Операнды A, B в последовательном коде, начиная со старших разрядов, поступают одновременно на входы LA и LB и через однотактные приемные регистры PA и PB подаются на одноразрядное АЛУ. С выхода АЛУ результат S может поступать на выход LS через однотактный выходной регистр PS или через один или два шеститактных регистра задержки результата P_d. Пары операндов A, B могут следовать друг за другом через необходимые промежутки времени или потоком. Управление масштабированием и задержкой выдачи результата осуществляется на регистрах P_d и PS сигналами с БУО.

Блок умножения обеспечивает получение произведения суммы (разности) операндов C и D на третий операнд G. Операнды подаются на входы LC, LD, LG одновременно в форме последовательных кодов старшими разрядами вперед и могут поступать непрерывным потоком или с необходимыми промежутками. БУ работает также по конвейерному принципу. Операнды C и D проходят однотактные регистры PC и PD и суммируются (вычитаются) на одноразрядном сумматоре См. Операнд G через четырехтактный регистр PG поступает на вход умножителя У одновременно с результатом из сумматора $(C \pm D)$. Произведение $(C \pm D)G$ с умножителя поступает на выход LP через однотактный регистр PP. Масштабирование результата, если это необходимо по команде, осуществляется на умножителе и регистре PG с помощью соответствующих сигналов управления, формируемых в БУО.

Блок управления операциями по заданному коду команды F вырабатывает сигналы управления для всех узлов и устройств МП. Шестиразрядный код F подается на шины LF и по сигналу WF записывается в регистр операции PO. Дешифратор ДШ расшифровывает состояние PO в необходимый набор сигналов. Настройка и перестройка МП производится за один такт и может осуществляться в короткие промежутки длительностью один-два такта в потоке операндов.

Цифровая информация на входах и выходах МП представляется в двоичной знакоразрядной системе счисления с фиксированной запятой. Любое число X в этой системе записывается с использованием положительных и отрицательных двоичных цифр $+1=1$, $-1=\bar{1}$. Например, число 0,101 может быть записано следующим образом: $0{,}101=0{,}11\bar{1}$. Знак числа определяется по старшей значащей цифре. Так, число $0{,}00\bar{1}01$ в обычной записи имеет знак минус: $0{,}00\bar{1}01=-0{,}00011$.

Перевод из обычной двоичной системы в знакоразрядную происходит следующим образом:

если $X \geqslant 0$, то число записывается без изменения:

$$X=0{,}0101 \to X=0{,}0101;$$

если $X<0$, то число также не изменяется, только знак переносится на все значащие цифры:

$$X=-0{,}0101 \to X=0{,}0\bar{1}0\bar{1}.$$

Обратный перевод осуществляется сложением положительной и отрицательной частей знакоразрядного числа.

Каждый информационный вход и выход БИС МП LX, X=A, B, C, D, G, S, P состоит из двух шин — положительной LX^+ и отрицательной LX^-. Положительная часть числа передается по положительной шине, отрицательная — по отрицательной, а запятая — по обеим шинам:

Моменты (такты)	i	1 2 3 4 5 6 7 8
Число	X	, 1 0 $\bar{1}$ 1 0 1 $\bar{1}$
Состояния шины	LX^+	1 1 0 0 1 0 1 0
Состояния шины	LX^-	1 0 0 1 0 0 0 1

Запятая в микропроцессоре выполняет роль «маркера», по которому осуществляется начальная установка отдельных узлов и устройств и запускается операция умножения.

Сложение (вычитание) чисел X и Y в АЛУ и См производится с учетом записи операндов X, Y и результата Z в форме $X=X_1X_2\ldots X_n$, $Y=Y_1Y_2\ldots Y_n$, $Z=Z_1Z_2\ldots Z_n$, $Z=X \pm Y$ по алгоритму

$$S^i=2(S^{i-1}-4Z_{i-3})+X_i \pm Y_i,$$

$$Z_{i-2}=\begin{cases}1, & \text{если } S^i \geqslant 3,\\ 0, & \text{если } |S^i|<3,\\ \bar{1}, & \text{если } S^i \leqslant -3,\end{cases}$$

где $i=1, n+2$, $S^0=0$, $Z_{-1}=0$, Z_0 — цифра переполнения.

Пример. $Z=X+Y=0{,}1\bar{1}0\bar{1}$ 1 $0{,}1\bar{1}1\bar{1}\bar{1}$:

Номер цифры	i	0 1 2 3 4 5 6 7
Операнд	X	, 1 $\bar{1}$ 0 $\bar{1}$ 0
Операнд	Y	, 1 $\bar{1}$ 1 $\bar{1}$ $\bar{1}$
Остаток	S	0 2 2 5 0 $\bar{1}$ $\bar{2}$ $\bar{4}$
Результат	Z	, 1 0 0 0 $\bar{1}$

Алгоритмы вычислений в умножителе можно представить в следующей форме:

$$S^i=2\ (S^{i-1}-4Z_{i-4})+2^{-1}\ (X_iY^i+Y_iX^{i-1}),$$

$$Z_{i-3}=\begin{cases}1, & \text{если } [S^i] \geqslant 3,\\ 0, & \text{если } -3<[S^i]<3,\\ \bar{1}, & \text{если } [S^i] \leqslant -3,\end{cases}$$

где $i=\overline{1,\ 2n+3}$, $S^0=0$, $Z_{-3}=Z_{-2}=Z_{-1}=0$, Z_0 — цифра переполнения, $[S^i]$ — целая часть остатка, $X^{i-1}=X_1X_2\ldots X_{i-1}$, $Y^i=Y_1Y_2\ldots Y_i$.

Сложение многоразрядных чисел $2(S^{i-1}-4Z_{i-4}) \times X_iY^i$, Y_iX^{i-1} осуществляется на параллельных знакоразрядных сумматорах, переносы в которых распространяются не более чем на два разряда. Поэтому время цикла всего МП определяется только межрегистровыми передачами, которые имеют глубину не более шести логических вентилей.

В цифровой обработке сигналов в основном используются операции быстрого преобразования Фурье, нерекурсивного и рекурсивного фильтров, причем более эффективными являются комплексные фильтры, т. е. фильтры комплексных сигналов. Операции БПФ, НРФ и РФ имеют почти одинаковую структуру — основу их алгоритма составляют умножение и сложение комплексных чисел. Для осуществления операций сложения (вычитания)

$$\dot{Z}=\dot{X} \pm \dot{Y}$$

$$\mathrm{Re}\dot{Z}+j\mathrm{Im}\dot{Z}=\mathrm{Re}\dot{X} \pm \mathrm{Re}\dot{Y}+j(\mathrm{Im}\dot{X} \pm \mathrm{Im}\dot{X})$$

и умножения комплексных чисел

$$\dot{W}=\dot{U}\dot{V}$$

$$\mathrm{Re}\dot{W}+j\mathrm{Im}\dot{W}=\mathrm{Re}\dot{U}\cdot\mathrm{Re}\dot{V}-\mathrm{Im}\dot{U}\cdot\mathrm{Im}\dot{V}+$$

$$+\ j(\mathrm{Re}\dot{U}\mathrm{Im}\dot{V}+\mathrm{Im}\dot{U}\mathrm{Re}\dot{V})$$

необходимо объединить четыре микропроцессора МП1... МП4 в один, как показано на рис. 2.

На БАЛО МП1 и МП2 образуется сумма комплексных чисел. На БУ МП1 и МП2 — произведения соответственно действительных и мнимых частей комплексных сомножителей, а на БУ МП3 и МП4 — произведения действительной части на мнимую соответственно первого — второго и второго — первого сомножителей. Вычитание первой пары произведений осуществляется на БАЛО МП3 — действительная часть комплексного произведения. Сложение второй пары произведений осуществляется на БАЛО МП4 — мнимая часть комплексного результата

Аналогично строится процессор (рис. 3), реализующий базовую операцию БПФ («бабочку») с прореживанием по частоте 2:

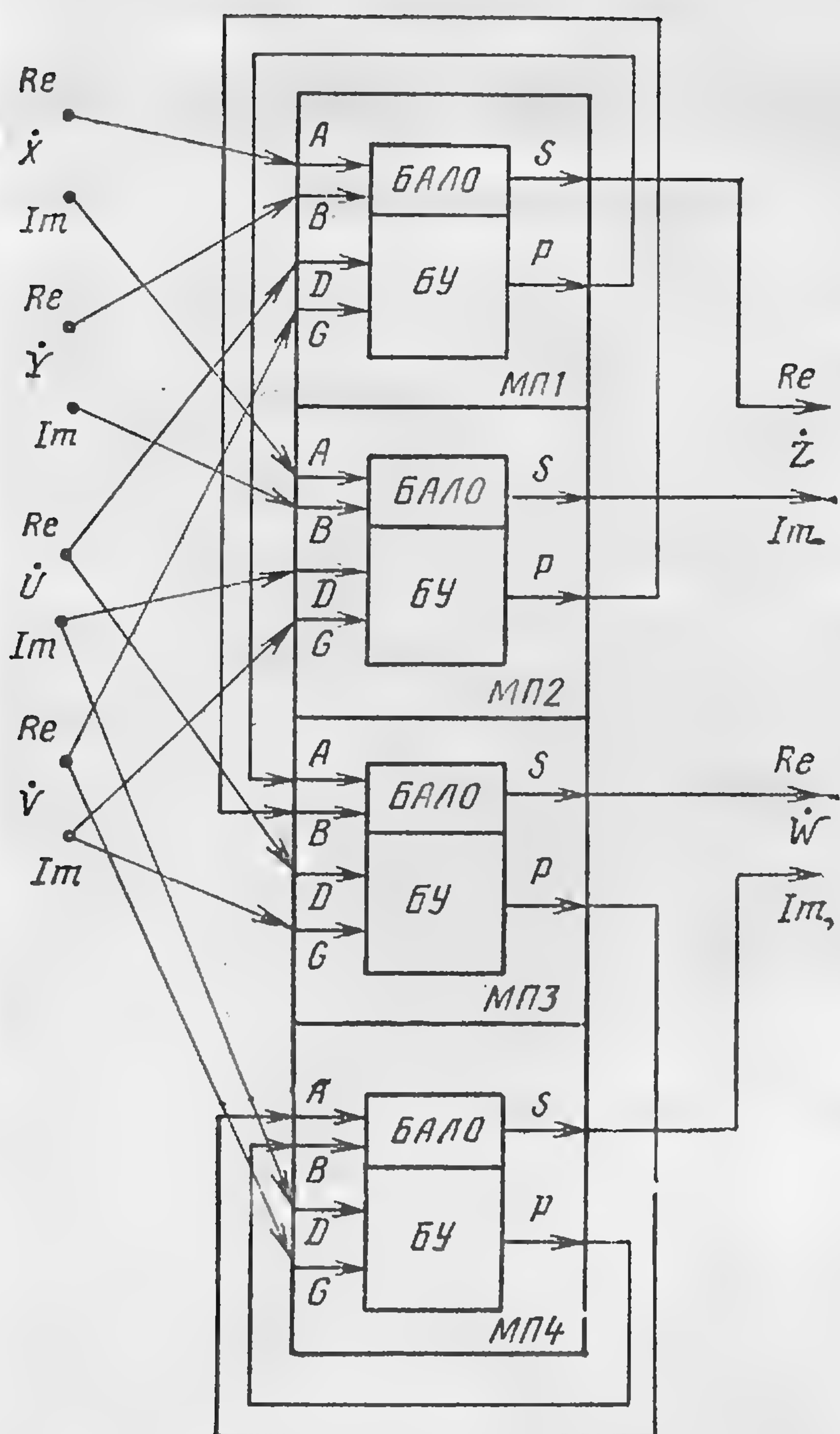


Рис. 2. Схема объединения четырех МП в процессор для умножения и сложения комплексных чисел

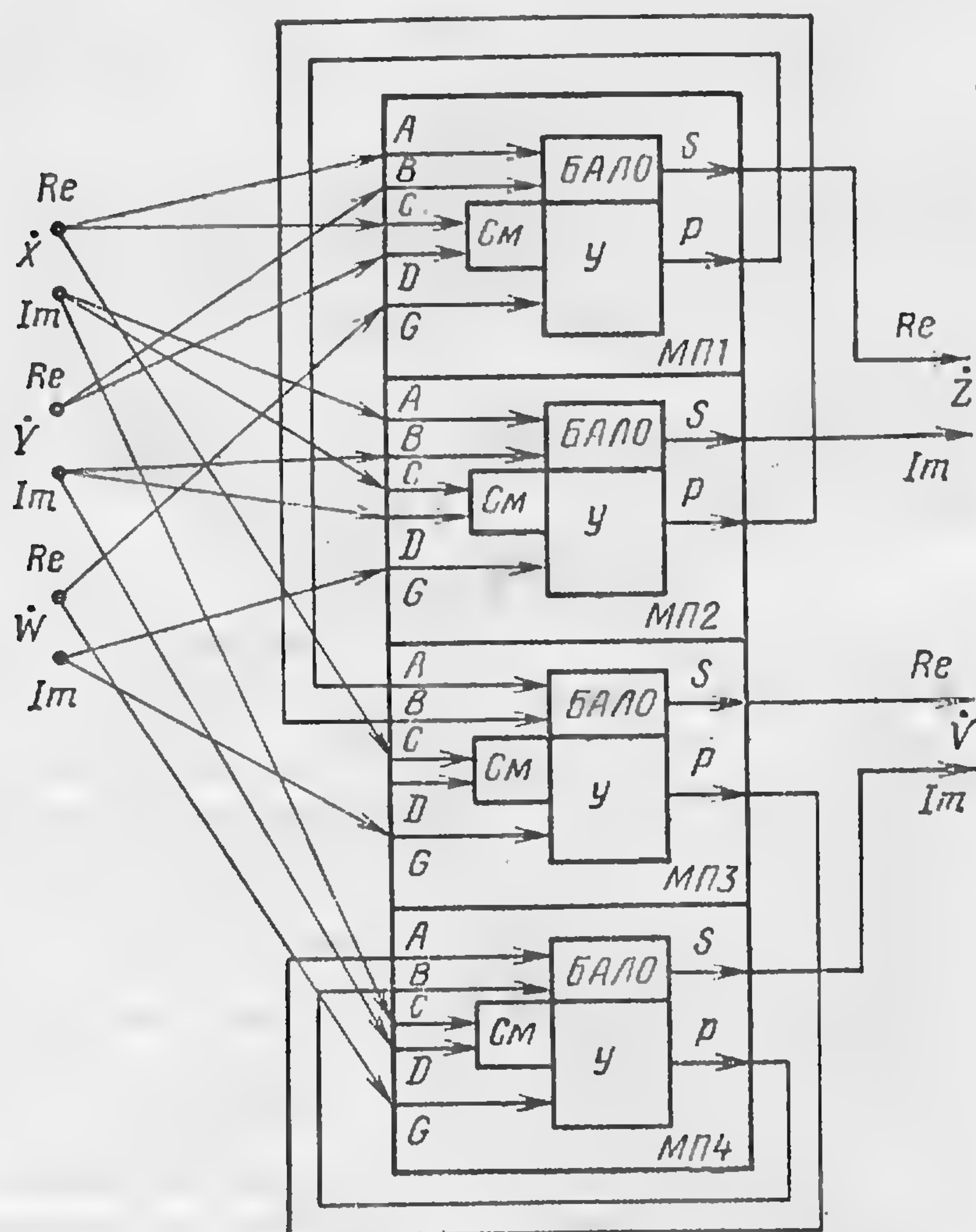


Рис. 3. Схема объединения четырех МП в процессор БПФ

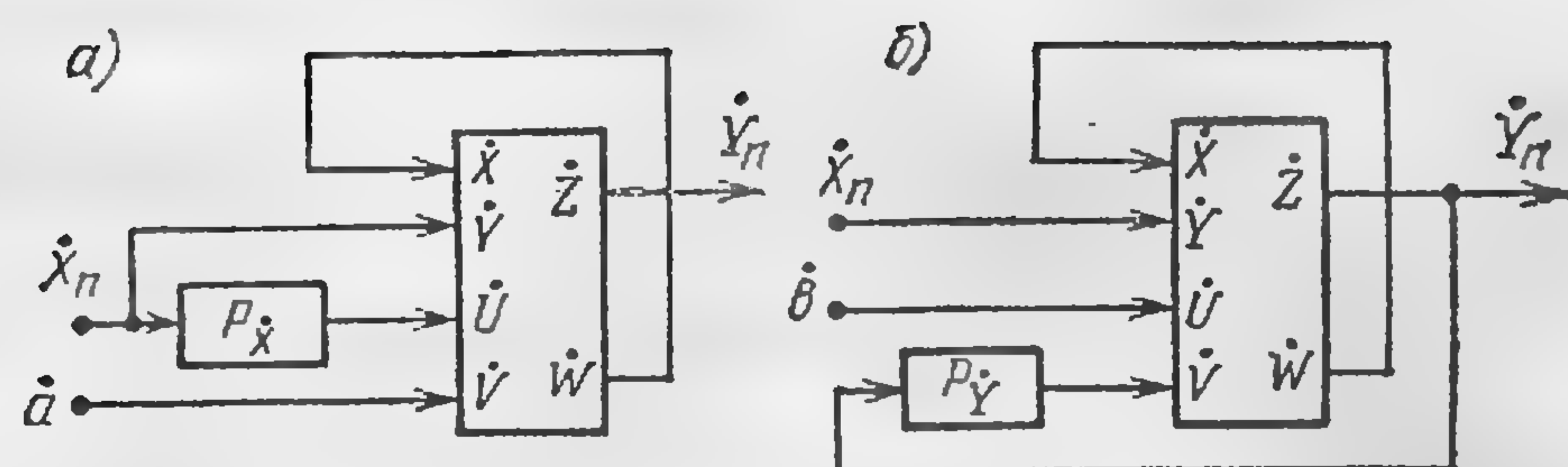


Рис. 4. Схемы комплексных фильтров: НРФ (а) и РФ (б)

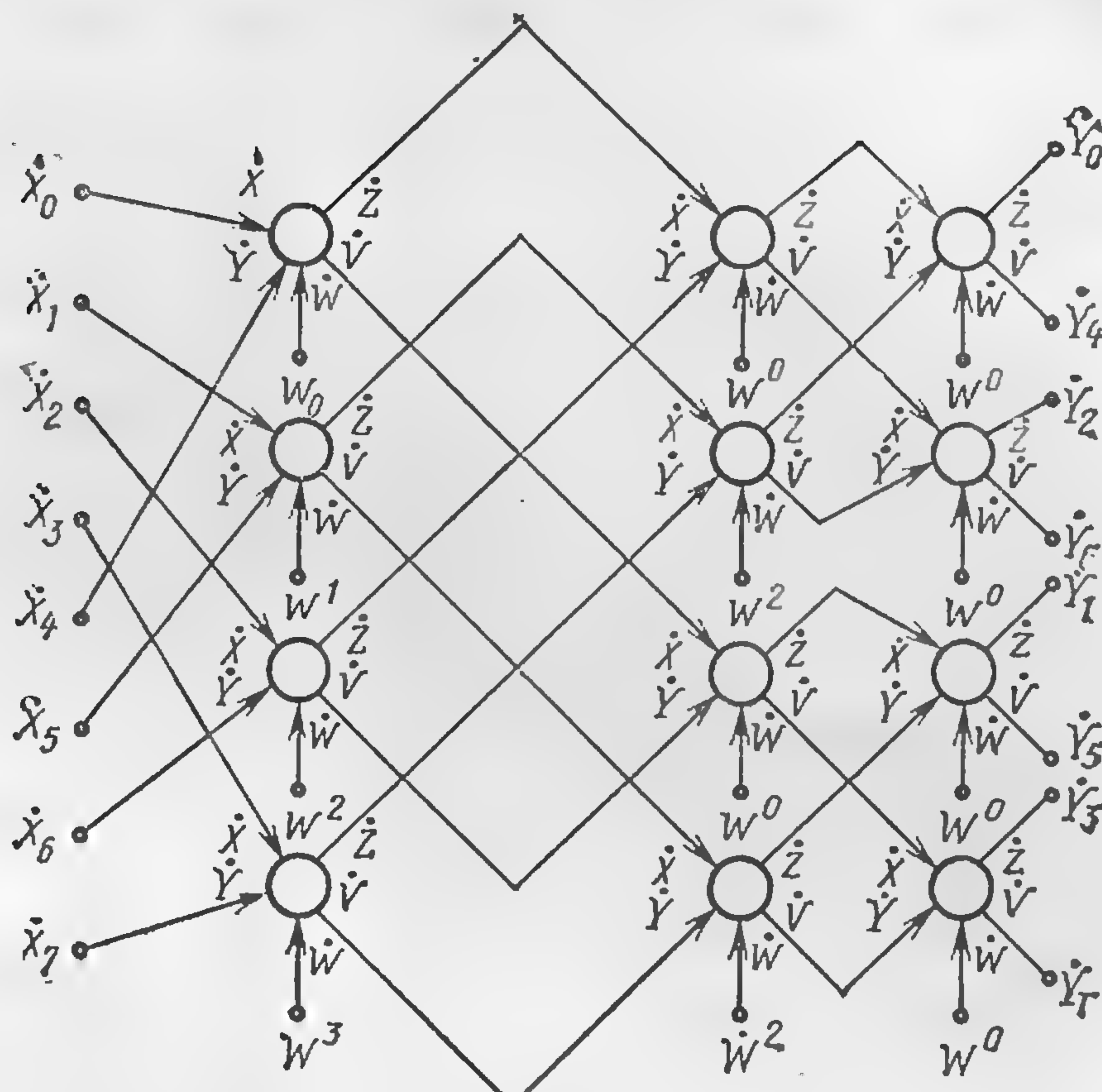


Рис. 5. Схема устройства восьмиточечного быстрого преобразования Фурье на основе процессоров БПФ

$$\dot{Z} = \mathrm{Re}\dot{Z} + j\mathrm{Im}\dot{Z} = \dot{X} + \dot{Y} =$$

$$= \mathrm{Re}\dot{X} + \mathrm{Re}\dot{Y} + j(\mathrm{Im}\dot{X} + \mathrm{Im}\dot{Y}),$$

$$\dot{V} = \mathrm{Re}\dot{V} + j\mathrm{Im}\dot{V} = (\dot{X} - \dot{Y})\dot{W} =$$

$$= (\mathrm{Re}\dot{X} - \mathrm{Re}\dot{Y})\mathrm{Re}\dot{W} - (\mathrm{Im}\dot{X} - \mathrm{Im}\dot{Y})\mathrm{Im}\dot{W} +$$

$$+ j[(\mathrm{Re}\dot{X} - \mathrm{Re}\dot{Y})\mathrm{Im}\dot{W} + (\mathrm{Im}\dot{X} - \mathrm{Im}\dot{Y})\mathrm{Re}\dot{W}].$$

Незначительное отличие состоит в том, что в сумматорах БУ дополнительно вычисляется разность преобразуемых отсчетов $\dot{X}$, $\dot{Y}$.

Процессор комплексного сложения и умножения позволяет достаточно просто получать нерекурсивные и рекурсивные фильтры. На рис. 4 показаны схемы комплексных НРФ (а) и РФ (б) первого порядка. Нерекурсивный фильтр описывается уравнением

$$\dot{Y}_n = \dot{X}_n + a\dot{X}_{n-1},$$

а рекурсивный — уравнением

$$\dot{Y}_n = \dot{X}_n + b\dot{X}_{n-1},$$

где X_n, $n=0, 1, 2, \ldots,$ — входная последовательность отсчетов комплексного сигнала; Y_n, $n=0, 1, 2, \ldots,$ — выходная комплексная последовательность реакции фильтра; a, b — коэффициенты фильтров.

Продолжение см. на стр. 85.

УДК 681.322.1

Д. А. Тилинин

ПЕРСОНАЛЬНАЯ ЭВМ «ОКЕАН 240»

Персональная ЭВМ является незаменимым инструментом исследователя, занимающегося сбором первичных данных, если она имеет интерфейс для сопряжения с измерительной аппаратурой и, таким образом, может быть использована для предварительной обработки и наглядного представления информации о результатах измерений. Однако применение традиционных ПЭВМ в жестких эксплуатационных условиях, например в морской экспедиции, для которой характерны повышенная влажность, вибрации и резкие изменения температуры воздуха, ограничено по ряду причин. Персональные ЭВМ, как правило, не имеют интерфейса для связи с нестандартной периферийной аппаратурой — задача сопряжения решается с помощью специализированных модулей, связанных с общей шиной ЭВМ. При всех своих преимуществах такой подход имеет существенный недостаток — повышенная вероятность отказа связей между модулями в сложных эксплуатационных условиях приводит к снижению надежности. Одним из путей существенного повышения надежности микроЭВМ является переход к одноплатной конструкции, которая позволяет значительно сократить число разъемных соединений, являющихся основной причиной отказов, и исключить магистральные усилители, на долю которых приходится значительная часть потребляемой электрической мощности.

Персональная ЭВМ «Океан 240» предназначена для работы в экспедиционных условиях. При ее разработке преследовалась цель создать простую и надежную в эксплуатации микроЭВМ с относительно низким энергопотреблением. В основу ПЭВМ «Океан 240» положен распространенный микропроцессорный комплект БИС серии К580. Оперативное ЗУ реализовано на микросхемах К565РУ5, постоянное репрограммируемое ЗУ на БИС К573РФ4. Все устройства, объединенные системной шиной (ЦП, ОЗУ, ПЗУ, контроллеры УВВ), расположены на одной печатной плате. Простота сопряжения с нестандартной периферийной аппаратурой обеспечивается набором программируемых устройств параллельного и последовательного обмена. Основные технические характеристики ПЭВМ «Океан 240» приведены ниже.

Персональная ЭВМ «Океан 240» выполнена по классической структурной схеме с 8-разрядной шиной данных и 16-разрядной шиной адреса (рис. 1). Центральный процессор К580ВМ80 связан системной шиной с ЗУ, РПЗУ и устройствами ввода-вывода: четыре БИС параллельного интерфейса К580ИК55, БИС последовательного обмена К580ВВ51, программируемый таймер К580ВИ53, контроллер приоритетных прерываний К580ВН59. Сигналы управления обменом по шине вырабатываются БИС контроллера К580ВК28, частота тактирования ЦП равна 2,4 МГц. Время цикла ОЗУ — 400 нс. Устройство управления ОЗУ (УОЗУ) выполняет функции регенерации и отображения области ОЗУ на экран ТВ-монитора в виде растра 256×256 точек в цветовом режиме и 512×256 точек в монохромном. Синхронный принцип обмена по шине ЦП К580ВМ80 позволяет минимальными средствами сделать видео-ЗУ «прозрачным» для ЦП, т. е. осуществлять вывод информации на экран в течение свободных тактов шины. Структурная схема и временные диаграммы работы устройства, синхронизирующего работу УОЗУ и ЦП, приведены на рис. 2. Входными для устройства синхронизации (УС) являются сигналы VR (разрешение записи информации в регистры видеоконтроллера), информация о состоянии ЦП S1, S4, S6, передаваемая по шине данных в такте T1, сигнал SYNC, обозначающий начало цикла передачи данных по шине ЦП, и сигнал RAM (выборка ОЗУ). На основании этих сигналов УС вырабатывает сигналы R1, R3, R4, управляющие работой УОЗУ. Регистр сдвига RG тактируется сигналом Ф2 (на рисунках соответствует обозначение 02). Таким образом, все изменения состояния схемы происходят по спаду Ф2.

В отсутствие сигнала SYNC, при появлении запроса от УОЗУ на считывание информации, схема вырабатывает импульс R4 длительностью в один такт Ф2, сбрасывающий триггер запроса и разрешающий чтение ОЗУ. При этом данные, считываемые из ОЗУ, записываются в регистры видеоконтроллера по спаду сигнала R4. При появлении сигнала SYNC, означающего начало цикла обмена ЦП с ОЗУ, на входах регистра D, D1 появляется сигнал «Лог.1», который записывается в регистр параллельно либо последовательно в зависимости от состояния шины S1 в такте T1, определяющей тип цикла обмена (запись — чтение).

В цикле чтения сигнала R1, разрешающий обмен с ЦП, вырабатывается в такте T2, а в цикле записи — в такте T3, что необходимо для обеспечения временных соотношений при передаче данных по шине ЦП.

В цикле записи сигнал R1 сопровождается сигналом R3, который используется для разрешения записи в ОЗУ. Появление сигнала R1 блокирует запрос от ОЗУ и таким образом запрещает одновременный доступ к ОЗУ двух активных устройств (ЦП и УОЗУ), причем ЦП имеет более высокий приоритет. Сигналы R1, R4 используются для управления адресными мультиплексорами и схемой формирования стробов $\overline{RAS}$, $\overline{CAS}$.

Разрядность, бит	8
Быстродействие (рег. — рег.), операций/с	600 000
Емкость ОЗУ, К байт	128
Емкость РПЗУ, К байт	16
Монитор (серийный цветной):	
в монохромном режиме	20 строк по 64 символа
в цветном режиме (4 цвета)	графическое поле 256×256
Клавиатура алфавитно-цифровая с контроллером на КМОП ИС	
Интерфейс:	
последовательный	RS 232-C
три параллельных программируемых канала	К580ИК55
интерфейс печатающего устройства	ИРПР

Программное обеспечение: монитор с графическими функциями, интерпретатор Бейсик, квазидисковая операционная система, совместимая с CP/M

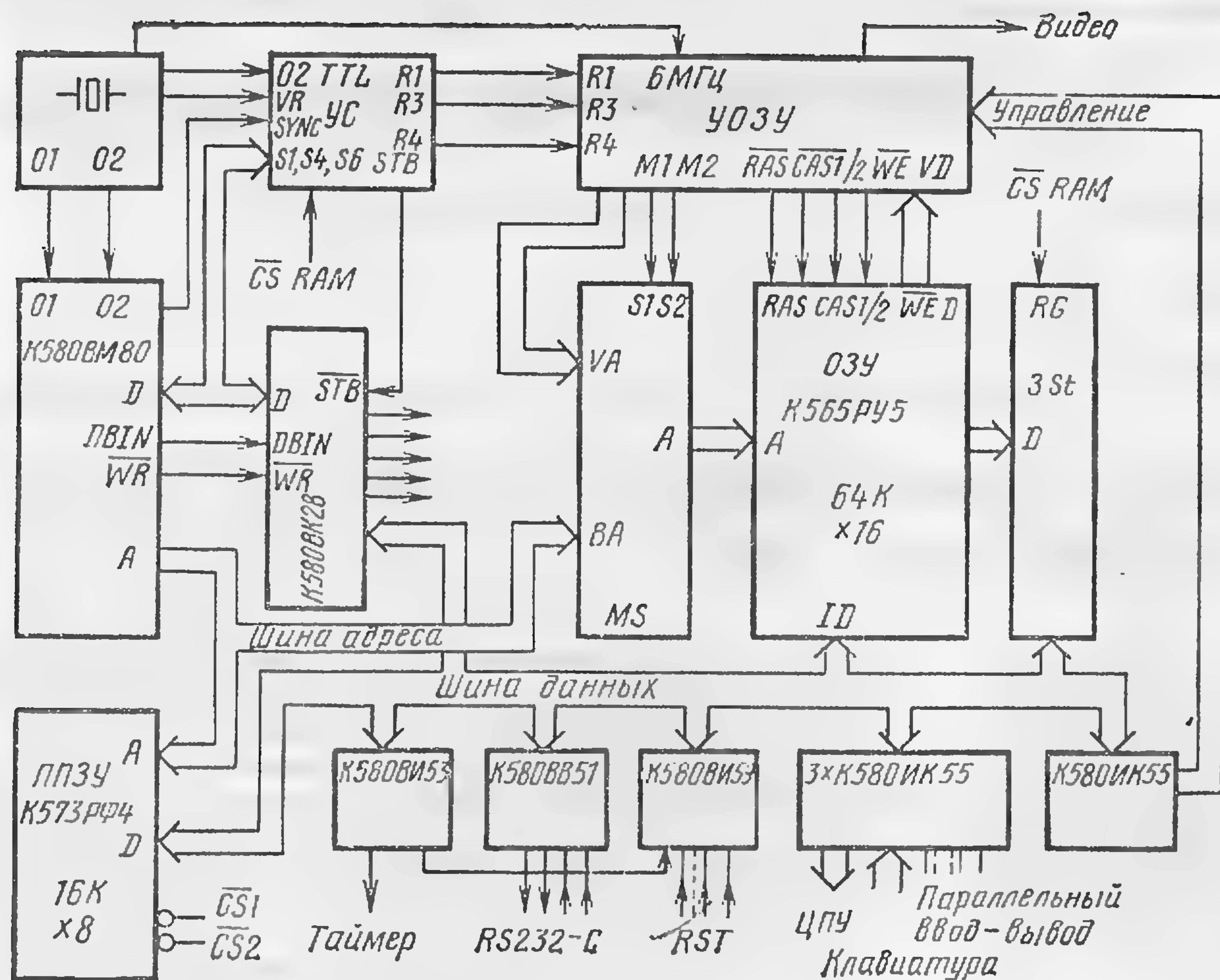
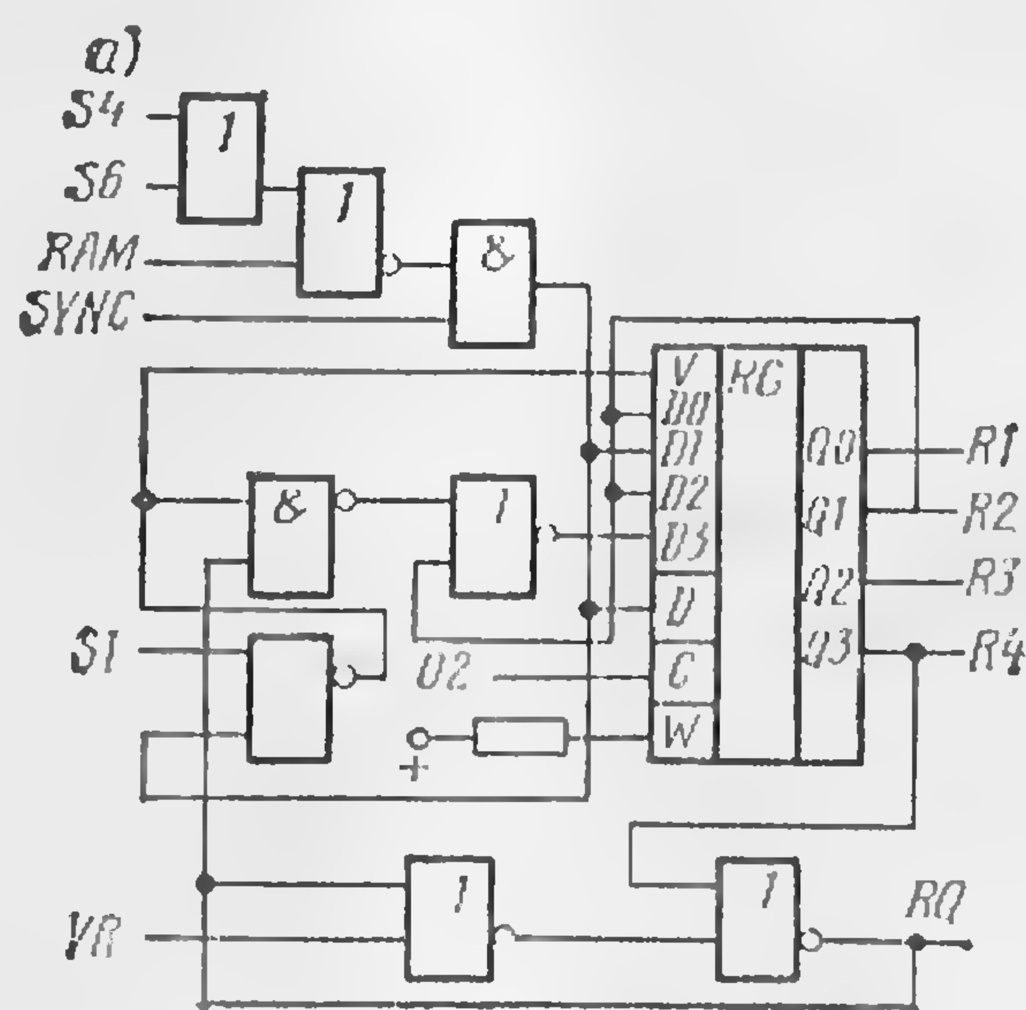


Рис. 1. Структурная схема персональной ЭВМ «Океан 240»

В цикле обмена с ЦП сигналы считывания и записи ОЗУ вырабатываются в адресном пространстве 0 ... 0BFFFH — в старших 16 К байт, соответствующих видео-ЗУ (адрес 0C000H ... 0FFFFH), располагается ППЗУ. При обмене с видео-ЗУ ЦП переключает его адрес с помощью регистра адресации, в качестве которого использован один из каналов БИС параллельного интерфейса К580ИК55. Регистр адресации выполняет также функции управления режимом отображения и доступом к дополнительному ОЗУ (64К байт), используемому в качестве «электронного диска» в операционной системе ПЭВМ.

Квазидисковая операционная система, совместимая с распространенной ОС CP/M, имеет объем ОЗУ пользователя 48К байт. Интерфейс с графическим ТВ-дисплеем обеспечивается программой «Монитор 240», в функции которой входят: управление режимами отображения, синтез алфавитно-цифровой информации, реализация графических функций LINE, SQUARE, SYMBOL, управление перемещением курсора, диагностика работы ПЭВМ. Кроме того, в составе «Монитора 240» содержатся драйверы связи с системными устройствами ввода-вывода: последовательной линией RS232-C, печатающим устройством типа УВВПЧ-30-004, кассетным магнитофоном, алфавитно-цифровой клавиатурой. Операционная система и монитор являются резидентными программами, т. е. физически записаны в ППЗУ, причем Console Command Processor (CCP) операционной системы загружается в старшие адреса ОЗУ для обеспечения совместимости с программами, использующими область CCP для стека (например, FORTRAN-80 Microsoft). Дополнительное ОЗУ разделено на два «электронных диска» (32К байт). Предусмотрено использование БИС динамического ОЗУ емкостью 256К бит (объем «дисков» может быть увеличен до 128К байт.)



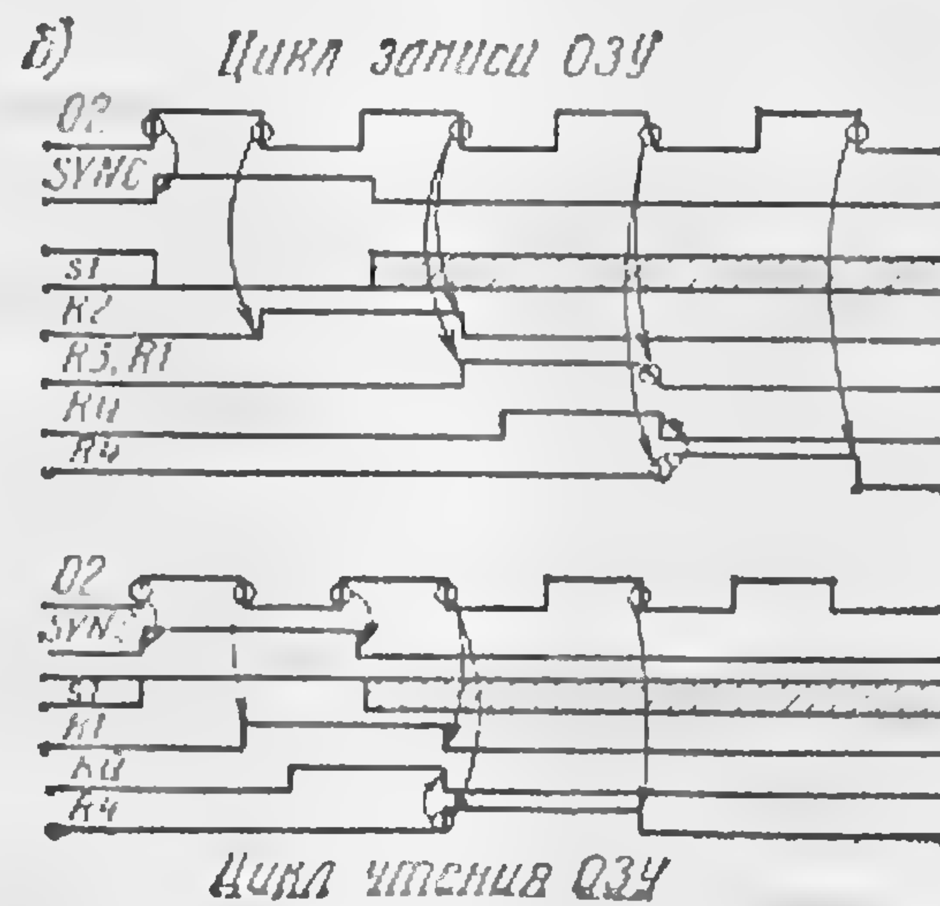


Рис. 2. Структурная схема (а) и временные диаграммы работы (б) устройства синхронизации

Одним из примеров применения ПЭВМ «Океан 240» может служить ИВК для измерения и экспресс-обработки гидрологических параметров, разработанный в Институте океанологии АН СССР. В состав ИВК кроме ПЭВМ входит глубоководный зонд с набором датчиков, измеряющих значения давления P, температуры t, электропроводности S и концентрации кислорода X. Первичные данные после аналого-цифрового преобразования передаются по кабель-тросу в устройство сопряжения, размещенное в корпусе микроЭВМ. Программа обработки, написанная на языке Фортран, выполняет следующие действия: вычисление абсолютных значений глубины, температуры, солености, относительной плотности и содержания кислорода с учетом взаимного влияния параметров и нелинейности датчиков; отображение графиков зависимостей параметров от глубины на экране ТВ-монитора; накопление измеряемых значений в буфере ОЗУ для вывода на кассетный НМЛ; распечатка абсолютных значений параметров при наличии печатающего устройства. Такой ИВК при всей его простоте, малых габаритах и относительно низком энергопотреблении (около 15 Вт) заменяет по своим функциям судовой зондирующий комплекс на базе судовой миниЭВМ типа ЕС-1010. Более того, возможность автономного питания от аккумуляторных батарей позволяет применить его при проведении гидрологических исследований с борта малых судов (катеров, шлюпок и т. д.), не имеющих мощной бортовой электросети. Эксплуатация ИВК в морских экспедициях на судах института океанологии подтвердила его высокую надежность и практичность.

Очевидно, что область применения ПЭВМ «Океан 240» не ограничена задачами первичной обработки информации по жесткому алгоритму — наличие ОС, совместимой с CP/M, позволяет применить обширный арсенал прикладных программ, написанных для CP/M-80: трансляторы, текстовые редакторы.

Адрес для запроса дополнительной информации: 117218, Москва, ул. Красикова, 23, Институт океанологии АН СССР.

Статья поступила 12 декабря 1985 г.

УДК 681.3.06

В. П. Телух, Г. В. Лебедева, Н. Л. Никитина,
Г. И. Пухальская

РАЗМЕЩАЕМЫЙ В ПЗУ ПАКЕТ СТАНДАРТНЫХ ПРОГРАММ ДЛЯ МИКРОЭВМ «ЭЛЕКТРОНИКА 60» И ДВК

Сфера применений микропроцессорной техники настолько широка, что возникает проблема ее обеспечения программными средствами, наиболее адекватными классам решаемых задач. Дисковые конфигурации микроЭВМ «Электроника 60» поддерживаются поставляемой в составе комплекса операционной системой (ОС) ФОДОС, которая обеспечивает широкий круг применений. В ряде применений, особенно при сложных условиях эксплуатации, целесообразно использовать бездисковые конфигурации микроЭВМ с ПЗУ и/или ППЗУ. Использование ФОДОС для подобных конфигураций невозможно [1].

В настоящее время создана исполнительная операционная система реального времени, пригодная для работы при размещении в ПЗУ [2]. Данная ОС, однако, не обеспечивает пользователей вычислительными ресурсами по преобразованию данных и вычислению элементарных функций, что осложняет разработку прикладных программных комплексов. Поставляемый в составе перфоленточной ОС микроЭВМ «Электроника 60» пакет стандартных программ (ПСП) в принципе можно использовать для предоставления этих ресурсов, но недостатки ПСП снижают эффективность его применения.

Невозможность выделения из поставляемого ПСП [2] компонент для самостоятельного использования приводит к неэффективному использованию внутренней памяти. Программы ПСП не обладают свойством реентерабельности для основного режима вызова, что обусловливается использованием для хранения результата зарезервированных ячеек ОЗУ (плавающий сумматор). В силу этого же ПСП [2] нельзя использовать при размещении в ПЗУ. Кроме того, время исполнения программ вычисления элементарных функций сильно зависит от значения аргумента и меняется от единиц до сотен миллисекунд, что затрудняет их использование в системах реального времени.

Учитывая ориентацию на применение в системах реального времени на базе бездисковых конфигураций микроЭВМ и недостатки имеющегося аналога, при разработке ПСП контролировалось выполнение требований к качеству программного изделия: точности, надежности, эффективности использования оперативной памяти и времени процессора, расширяемости состава и доступности отдельных компонент пакета для самостоятельного использования.

К специальным характеристикам, расширяющим сферу применения пакета и учтенным при его разработке, относятся реентерабельность, позиционная независимость и возможность использования при размещении в ПЗУ. Реентерабельность позволяет использовать весь пакет и отдельные его компоненты в качестве разделяемого между несколькими задачами вычислительного ресурса (общий объем используемой внутренней памяти сокращается). Позиционная независимость предоставляет дополнительную гибкость при размещении программ в памяти и их оперативном перемещении. Размещение и исполнение в ПЗУ повышает технологичность разрабатываемых систем за счет исключения из их состава перфоленточных устройств ввода.

Отметим, что часть из учитываемых при разработке характеристик качества, например точность и эффективность использования памяти и времени процессора, находятся между собой в обратной зависимости (оказался необходимым поиск компромиссных решений). В связи с этим на начальной стадии разработки пакета были проанализированы методы и алгоритмы вычисления элементарных функций для выбора наиболее пригодных с точки зрения предъявляемых требований к точности и эффективности. Проанализированы методы аппроксимации элементарных функций цепными дробями, рациональными и многочленными приближениями, разложением в ряд Тейлора; выбраны приближения с наименьшим количеством операций плавающей арифметики и с точностью не менее семи десятичных цифр. Отобранные алгоритмы промоделированы на языке Бейсик для подтверждения заданных требований к точности.

Структура пакета разработана с учетом требований его расширяемости и доступности компонент. На верхнем уровне пакета находится модуль идентификации запросов на стандартные программы (МИЗ) — головной. На нижнем уровне — модули стандартных программ. Подобное разбиение позволяет использовать пакет как единое целое компоновкой модуля МИЗ с модулями стандартных программ, а также использовать каждую из стандартных программ как самостоятельную единицу. При использовании ПСП как единого целого к стандартным программам можно обращаться без глобальных имен, применяя команду TRAP, младший байт которой соответствует коду вызываемой программы.

Функции головного модуля МИЗ — сохранение программного контекста, определение корректности запроса на стандартную программу, передача управления и параметров вызываемой стандартной программе, прием результатов и формирование кодов завершения. При табличной организации вызовов стандартных программ из модуля МИЗ состав компонент пакета легко расширить.

Времена исполнения стандартных программ (табл. 1) получены по результатам испытания пакета с использованием тест-программы (см. далее) и включают время на исполнение головного модуля МИЗ (около 0.2 мс). Времена исполнения стандартных программ нахождения минимальных и максимальных значений элементов массива зависят от параметра n, равного длине массива. Времена исполнения стандартных программ вычисления элементарных функций практически не зависят от значения аргумента. Объем памяти, требуемый для развития программ при последовательном обращении к ним, не превышает максимального объема памяти стека одной из вызываемых программ.

Таблица 1

Основные сведения о компонентах ПСП

Наименование программы	Имя программы	Число строк языка ассемблер	Объем памяти, байт	Требуемый объем стека, байт	Время исполнения, мс
Преобразование числа из символьного представления в вещественное	SWR	331	928	48	4,57
Преобразование вещественного числа в символьный вид	RWS	291	1124	48	4,93
Преобразование числа из символьного представления в целое	SWI	88	180	36	3,53
Преобразование целого числа в символьный вид	IWS	75	162	26	3,54
Преобразование вещественного в целое	RWI	91	174	10	4,23
Преобразование целого в вещественное	IWR	75	152	22	5,28
Вычисление элементарной функции «Синус»	SIN	153	478	44	9,73
Вычисление элементарной функции «Косинус»	COS	41	994	80	10,44
Вычисление элементарной функции «Тангенс»	TG	135	424	26	5,52
Вычисление элементарной функции «Экспонента»	EXP	126	324	50	14,24
Вычисление элементарной функции «Логарифм натуральный»	LN	99	280	80	5,27
Вычисление элементарной функции «Логарифм десятичный»	LOG	19	50	80	5,76
Вычисление корня квадратного	SQRT	92	250	26	9,45
Нахождение ближайшей целой части вещественного числа	BINT	94	212	12	2,26
Нахождение максимального значения элемента массива вещественных чисел	MAXR	81	170	24	$0{,}24+0{,}36\cdot n$
Нахождение максимального значения элемента массива целых чисел	MAXI	39	76	22	$0{,}24+0{,}03\cdot n$
Нахождение минимального значения элемента массива вещественных чисел	MINR	81	170	24	$0{,}24+0{,}36\cdot n$
Нахождение минимального значения элемента массива целых чисел	MINI	39	76	22	$0{,}24+0{,}03\cdot n$

Алгоритмы вычисления элементарных функций, включаемые в состав пакета, заимствованы из работ [3, 4].

Генератор целевых пакетов (ГЦП) — это специальное инструментальное программное средство, разработанное для облегчения сопровождения и тиражирования пакета стандартных программ. Основным назначением ГЦП является формирование целевого пакета стандартных программ из объектного пакета в соответствии с параметрами, задаваемыми оператором в режиме диалога. Целевой пакет — пакет для непосредственного использования в составе разрабатываемой системы и обладающий характеристиками, заданными оператором при его генерации. Объектный пакет — полный набор компонент, хранящихся в виде объектных модулей в дисковой памяти инструментальной ЭВМ.

Параметры генерации, задаваемые оператором,— это состав целевого пакета, вид представления пакета (загрузочный или объектные модули), носитель целевого пакета (перфолента, магнитная лента, диски, гибкие диски и т. п.), адрес точки входа (для загрузочного модуля). При генерации создается справочный файл с информацией, идентифицирующей пакет, а также ведется протокол генерации.

Таким образом, ГЦП позволяет получать пакеты стандартных программ разного состава на различных носителях в виде загрузочного или объектных модулей. ГЦП функционирует в среде ОС ФОДОС микроЭВМ «Электроника 60».

Специальная тест-программа, предназначенная для проверки соответствия ПСП требованиям технического задания, предусматривает определение времени исполнения стандартных программ и проверку:

корректности и надежности исполнения программ при вызове по команде TRAP и при вызове их как самостоятельных компонент по команде JSR;

расширяемости пакета;

точности вычисления элементарных функций во всем диапазоне представления аргументов.

Выполнение требований обеспечения реентерабельности и позиционной независимости проверялось по текстуальному представлению программ пакета. Позиционная независимость достигнута за счет использования позиционно-независимых режимов адресации, а свойство реентерабельности — выделением памяти в стеке под переменные при каждом вхождении в ПСП или отдельную компоненту.

Основные сведения о ПСП, ГЦП и тест-программе приведены в табл. 2.

Таблица 2

Основные сведения о ПСП, ГЦП и тест-программе

Наименование программы	Число операторов языка ассемблер	Требуемый объем памяти, байт
Пакет стандартных программ	2021	5 224
Генератор целевых пакетов	1180	19 326
Тест-программа ПСП	2288	27 010

Испытания пакета показали, что точность вычисления элементарных функций не ниже семи десятичных цифр, а время их исполнения на микроЭВМ «Электроника 60» с блоком плавающей арифметики не превышает 15 мс. Это значительно лучше временных характеристик пакета стандартных программ, входящего в состав ПЛОС.

Все программы ПСП разрабатывались на инструментальной ЭВМ в среде ФОДОС на подмножестве языка Макроассемблер. В настоящее время ПСП проходит опытную эксплуатацию. На него выпущена эксплуатационная документация, предусмотренная ЕСПД.

Благодаря доступности отдельных компонент, возможности выбора состава пакета, высоким временным ха-

рактеристикам, обеспечению реентерабельности и позиционной независимости, возможности размещения и исполнения в ПЗУ, легкости модификации разработанный ПСП можно по-разному применять в бездисковых конфигурациях микроЭВМ «Электроника 60».

Пакет стандартных программ сдан в фонд алгоритмов и программ ВНИГИК, г. Калинин (частичный фонд отраслевого фонда алгоритмов и программ Министерства геологии СССР). Пользователи могут получить исходные и объектные модули пакета, а также комплект эксплуатационной документации, включая алгоритмы всех стандартных программ.

Для получения пакета необходимо предоставить магнитный носитель для накопителей типа НГМД-70 или ИЗОТ-5003 или бумажную перфоленту и подготовить письмо на имя директора ВНИГИК: 170000, г. Калинин, ул. Урицкого, 37/60.

ЛИТЕРАТУРА

1. Телух В. П., Борисов Ю. И., Парфеньев Н. Д. Программное обеспечение автоматизированного рабочего места геофизика на базе микроЭВМ. — В кн.: Вопросы разработки и эксплуатации АСУ. — Калинин: Центрпрограммсистем, 1983, ч. 1, с. 211—215.

2. Зурахинский В. И. Операционная система реального времени для микроЭВМ. — Управляющие системы и машины, 1983, № 3, с. 52—54.

3. Система стандартных программ / Под ред. М. Шура-Бура. — М.: 1958.

4. Cody W. J., Waite W. Software manual for the elementary functions.— Englewood Cliffs: Prentice Hall, 1980.—269 p.

Статья поступила 5 декабря 1985 г.

УДК 681.3.06

А. В. Попурий

СИСТЕМА ПРОГРАММИРОВАНИЯ ДИАЛОГ ДЛЯ МИКРОПРОЦЕССОРА КР580ИК80

Создание программного обеспечения (ПО) микропроцессорных систем значительно упрощается, если использовать языки программирования высокого уровня. Однако известные системы программирования для микропроцессоров не удовлетворяют пользователей либо из-за отсутствия диалогового режима в разработке и отладке программ, либо из-за потребности в дорогостоящем оборудовании (например, при работе с кросс-средствами), либо из-за низкой скорости выполнения программ, как, например, в интерпретирующих системах. Учитывая специфику программирования микропроцессоров, предпочтение при выборе языка чаще всего отдается ассемблеру, позволяющему наиболее полно использовать систему команд и занимаемую память.

Отсутствие диалогового режима при создании программ проявляется в том, что пользователь как бы отделен от машины и не является хозяином положения: для опробования только что написанного небольшого участка программы приходится либо производить полную перетрансляцию всей программы, либо транслировать и вводить в память вручную так называемые заплаты, изменяющие содержимое тела программы. При этом возможно рассогласование исходного текста с объектным кодом из-за ошибок коррекции исходного текста после откладки. Нередки случаи, когда эксплуатируемая программа как бы «замораживается» и внести в нее изменения в случае выявления ошибки очень сложно.

Диалоговые языки и системы программирования ускоряют отладку и повышают качество рабочих программ, так как позволяют оперативно вносить изменения, и испробовав несколько вариантов, выбрать наилучший.

Из распространенных языков программирования можно выделить Форт [1] и Бейсик, прямо ориентированные на диалоговый режим откладки. При использовании пары Бейсик-интерпретатор и Бейсик-компилятор отладку производят, применяя интерпретатор, т. е. очень замедленно (существенный недостаток для программирования встроенного МП).

Диалоговый язык Форт при всей его необычности становится все более популярным благодаря исключительно удобной отладке, гибкости и наращиваемости.

Разработанная система программирования Диалог в части вычислительных средств языка практически полностью аналогична Форту и тоже использует для этой цели дополнительный программно-организованный стек (операндов) Отличия системы Диалог от системы Форт приведены в разделе описания базовых операторов.

Система Диалог сочетает преимущества компилирующих и диалоговых систем, требует минимального набора оборудования и памяти. В нее входят: однопроходный компилятор с языка высокого уровня Диалог, имеющего в своем составе полный набор команд ассемблера, встроенный редактор текстов и средства отладки. Система работает в режиме калькулятора, т. е. сразу же выполняет команды, введенные с терминала в символическом виде, а также позволяет вводить новые операторы языка, готовые для немедленного выполнения в реальном времени. В языке предусмотрены вычислительные средства, средства работы с аппаратурой, средства структурированного управления.

Система обеспечивает ввод и редактирование любого участка программы на уровне исходного текста, создание промежуточных отладочных вариантов программы с быстрой загрузкой, генерацию загрузочного файла, пригодного для записи в ПЗУ или для работы в операционной среде.

Система Диалог (6К байт) реализована для микропроцессора К580ИК80 и может быть реализована на других типах микропроцессоров и мини-ЭВМ.

Для работы системы необходим рабочий объем ОЗУ (не менее 3К байт), зависящий от объема разрабатываемой программы. Минимальный набор оборудования: фотосчитыватель, перфоратор и терминал. Эффективность системы повышается при использовании дисковых операционных систем.

Вычислительные средства языка

В языке имеется базовый набор операторов для вычислений с целыми 16-битными числами, однако пользователь может без особых затруднений дополнить этот набор собственными операторами для работы с числами любой длины. Для хранения операндов, промежуточных результатов вычислений и для передачи параметров между операторами используется программно-организованный стек, который не следует смешивать с аппаратным стеком, используемым обычно для организации подпрограмм.

Пример процесса вычислений. Пусть требуется вычислить и вывести на экран терминала результат выражения 65 * 23-198/3. Для этого необходимо исходное выражение ввести с терминала в обратной польской записи 65 23 * 198 3/-. На экране появится результат 1429. Последовательность действий при вычислении выражения запишем в виде таблицы:

ТЕРМ[illegible]	ДЕЙСТВИЯ ЭВМ	СОСТОЯНИЕ СТЕКА ПОСЛЕ ВЫПОЛНЕНИЯ ОПЕРАЦИИ
	ИСХОДНОЕ СОСТОЯНИЕ	
65	ПОМЕСТИТЬ В СТЕК ЧИСЛО 65	65
23	ПОМЕСТИТЬ В СТЕК ЧИСЛО 23	23 65
* (УМНОЖЕНИЕ)	ЗАМЕНИТЬ ДВА ВЕРХНИХ ЭЛЕМЕНТА ИХ ПРОИЗВЕДЕНИЕМ	1495
198	ПОМЕСТИТЬ В СТЕК ЧИСЛО 198	198 1495
3	ПОМЕСТИТЬ В СТЕК ЧИСЛО 3	3 198 1495
/ (ДЕЛЕНИЕ)	РАЗДЕЛИТЬ НА ВЕРХНИЙ ЭЛЕМЕНТ	66 1495
— (ВЫЧИТАНИЕ)	ВЫЧЕСТЬ ВЕРХНИЙ ЭЛЕМЕНТ	1429
. (ТОЧКА)	ВЫВЕСТИ РЕЗУЛЬТАТ НА ТЕРМИНАЛ	

В данном примере числа 65, 23, 198 и 3 — десятичные константы, а знаки «*», «/», «—», «.» — имена операторов (соответственно умножения, деления, вычитания и печати). Именем в языке Диалог считается цепочка любой длины, состоящая из букв и/или цифр, начинающаяся с буквы. Кроме того, в качестве имени может фигурировать знак. Ограничитель буквенно-цифрового имени — это любой символ, не являющийся буквой или цифрой, в том числе и имя-знак. Имена-знаки не требуют ограничителя и могут записываться подряд, как в описанном выше примере. Исключение составляют знаки «?» и «.» (точка), считающиеся буквами.

Константы в языке Диалог записываются, как в ассемблере для К580ИК80: десятичные — без каких-либо добавлений, восьмеричные — с добавлением буквы «Q» в конце константы, шестнадцатеричные — буквы «H», причем последние должны начинаться с десятичной цифры, в конце двоичных констант записывается буква «B»:

```
65535           десятичная запись
177777Q         восьмеричная
0FFFFH          шестнадцатеричная
1111111111B     двоичная
```

Приведенный выше пример демонстрирует использование системы в качестве постфиксного калькулятора для вычисления выражений, однако этот режим с успехом используется и для управления аппаратурой. Например, для выдачи управляющего сигнала в 5-й порт ввода-вывода достаточно ввести с терминала две команды ассемблера, и эта операция немедленно будет выполнена:

```
MVI A.1   OUT 5
```

Команды ассемблера можно записывать в одной строке и отделять друг от друга пробелами или табуляцией. Символ «;» (точка с запятой) обозначает комментарий до конца строки.

Операторы ветвления

Для организации ветвлений по условиям в языке Диалог применяются конструкции структурного программирования: IF...ENDIF, IF...ELSE...ENDIF, BEGIN...WHILE. Поскольку метки в языке отсутствуют, эту роль выполняют операторы ENDIF и BEGIN, которые не исполняются, а только указывают точку в программе, в которую осуществляется условный переход.

Для МП К580ИК80 инструкции ветвления вперед следующие: IFZ, IFNZ, IFC, IFNC, IFP, IFM, IFPO, IFPE (т.е. образованы из мнемоник команд ассемблера заменой буквы «J» на буквы «IF»), и организуют ветвления по признакам соответственно нуля, переноса, знака и четности, например:

```
TSTZ
 IFZ
     ; ЭТА ЧАСТЬ ПРОГРАММЫ ВЫПОЛНЯЕТСЯ, ЕСЛИ
       В СТЕКЕ
     ; БЫЛ НУЛЬ, ИНАЧЕ ОНА ПРОПУСКАЕТСЯ
     ......
ENDIF
```

Оператор TSTZ анализирует на нуль верхний элемент стека и удаляет его, устанавливая признак Z в соответствующее состояние. После выполнения участка программы

```
RRC
IFC
    20 ; ПОМЕСТИТЬ В СТЕК ЧИСЛО 20 И ПЕРЕЙТИ
         НА ENDIF
ELSE
     5
ENDIF
```

в стек будет помещено число 20 или 5 в зависимости от предшествующего состояния младшего бита сумматора (соответственно 1 или 0).

Расширяемость языка

Структура языка Диалог позволяет дополнять базовый набор операторов операторами пользователя и сразу же их использовать наряду с уже имеющимися в системе. Записав строку

```
<ПОЛУСУММА + 2/ >
```

пользователь определяет новый оператор с именем ПОЛУСУММА. Открывающая угловая скобка сообщает системе о введении нового или замене старого оператора, далее следует имя, затем тело оператора, которое заканчивается закрывающей угловой скобкой. Этот оператор вычисляет полусумму двух значений, ранее записанных в стек, и заменяет их одним числом. Использовать только что введенный оператор можно так:

```
5 14 ПОЛУСУММА.
```

На экране терминала появится число 9.

Пример оператора с возвратом по условию:

```
<ЗАДЕРЖКА
;
; ОПЕРАТОР ПРОГРАММНОЙ ЗАДЕРЖКИ ВРЕМЕНИ
;
MVI B,120        ; ЧИСЛО ПОВТОРЕНИЙ ЦИКЛА
BEGIN            ; ТОЧКА ВОЗВРАТА (НЕ ИСПОЛ-
                 ; НЯЕТСЯ)
    DCR B        ; ДЕКРЕМЕНТ И ПРОВЕРКА
                 ; СЧЕТЧИКА
WNZ              ; ВОЗВРАТ НА BEGIN, ЕСЛИ НЕ
                 ; НУЛЬ
>                ; КОНЕЦ ОПРЕДЕЛЕНИЯ ОПЕРА-
                 ; ТОРА
```

Операторы WZ, WNZ, WC, WNC, WP, WM, WPO, WPE, также образованы из мнемоник команд ассемблера заменой буквы «J» на букву «W» и служат для передачи управления назад в точку BEGIN.

Конструкции управления могут быть вложены одна в другую, однако перекрещивание их не допускается. Чтобы облегчить прослеживание выполнения программы, рекомендуется текст записывать в виде уступов, как это обычно делается в языках типа Алгол:

```
<ПРЕОБРАЗОВАНИЕ
;
; ШЕСТНАДЦАТЕРИЧНЫЙ КОД В СУММА-
ТОРЕ
; ПРЕОБРАЗУЕТСЯ В КОД КОИ-7
;
ANI OFH CPI 10
IFC
      ADI 30H
ELSE
      ADI 20H CPI 2FH
      IFZ
            MVI A20H
      ENDIF
ENDIF>
```

Переменные, массивы, загружаемые константы

Для присвоения имен переменным, рабочим областям и для создания различных таблиц применяются операторы описания VAR, ARRAY и CONST:

```
VAR ПЕРЕМ            ; В РАБОЧЕЙ ОБЛАСТИ РЕ-
                       ЗЕРВИРУЮТСЯ
                     ; 2 БАЙТА
ARRAY МАССИВ,16      ; В РАБОЧЕЙ ОБЛАСТИ РЕ-
                       ЗЕРВИРУЮТСЯ
                     ; 16 БАЙТ
CONST C1,5,1         ; В ОБЛАСТИ КОНСТАНТ
                       ЗАПОЛНЯЮТСЯ
                     ; 4 БАЙТА
CONST T1,'ТЕКСТ'     ; ЗАПОЛНЯЮТСЯ 5 БАЙТ
                       С ИМЕНЕМ «Т1»
```

Каждому имени, указанному в одном из операндов VAR, ARRAY или CONST, ставится в соответствие адрес локализации соответствующей области памяти. Запись этого имени в программе означает, что адрес локализации данного объекта необходимо поместить в стек.

Значение такого объекта вызывается оператором «@». Пусть, например, в рабочих ячейках МНОЖИМОЕ и МНОЖИТЕЛЬ находятся какие-либо значения. Для перемножения и индикации произведения на экране дисплея необходимо записать строку:

МНОЖИМОЕ @ МНОЖИТЕЛЬ @ *.

Операторы специального вида

Так как микропроцессоры чаще всего применяются для управления аппаратурой, введена специальная синтаксическая форма оператора со списком параметров неопределенной длины, содержащим пары шестнадцатеричных чисел, например:

УПР 5,C,8,0,3A,15

Хотя такая форма и нарушает синтаксическую структуру языка, однако она очень полезна в приложениях. Такой записи можно придать смысл команд управления: выдать в порт 05 код 0C, в порт 08 код 00, в порт 3A код 15. Характерно, что в языке существует только способ компиляции такого оператора, а сам оператор разрабатывается пользователем с учетом конкретных особенностей управляемой аппаратуры. Для объявления оператора специального вида применяется оператор описания SPECIAL. Такая форма позволяет лаконично описать процесс управления.

Таблица 1

Базовый набор операторов

Оператор	Форма записи	Выполнение
	Операторы описания	
<	<ИМЯ	Определить новый оператор с именем ИМЯ. Далее следует тело оператора
>	>	Конец определения оператора
SPECIAL	SPECIAL ИМЯ	Определить оператор специального вида
VAR	VAR ИМЯ1, ИМЯ2 ...	Определить двухбайтовые переменные ИМЯ1, ИМЯ2, ...
ARRAY	ARRAY ИМЯ, ДЛИНА	Объявить массив ИМЯ длиной ДЛИНА в байтах
CONST	CONST ИМЯ, X1, X2 ,...,	Определить одну или несколько загружаемых констант. Первому байту присвоить имя ИМЯ. X1, X2, ... — положительные или отрицательные двухбайтовые константы, имена (адресные константы) операторов, переменных, массивов, констант, а также текстовые константы
END	END	Конец программы
;	; ТЕКСТ	Комментарий до конца строки
	Операторы обработки данных	
+	X У +	Сложение
—	X У —	Вычитание, в стеке остается X—У
NEG	X NEG	Образование дополнительного кода X
*	X У *	Умножение
/	X У /	Деление, в стеке остается X/У, остаток отбрасывается
MOD	X У MOD	В стеке остается остаток от деления X на У
AND	X У AND	Логическое умножение
OR	X У OR	Логическое сложение
XOR	X У XOR	Исключающее ИЛИ
SHL	X У SHL	X логически сдвигается на У бит влево и остается в стеке
SHR	X У SHR	X логически сдвигается на У бит вправо
DUP	X DUP	Дублирование верхнего элемента стека
SWAP	X У SWAP	Перестановка двух верхних элементов стека
@	X @	X заменяется содержимым с адресом X
STORE	X У STORE	X записывается по адресу У
TSTZ	X TSTZ	X анализируется на нуль и удаляется из стека. Устанавливается признак Z в регистре F
TSTS	X TSTS	Анализ знака X (признак S)
TSTP	X TSTP	Анализ четности X (признак P)

Продолжение табл. 1

Оператор	Форма записи	Выполнение
COMP	X У COMP	Сравнение: У вычитается из X, устанавливается признак переноса СУ, в стеке остается X
.	X .	X печатается и удаляется из стека
CR	CR	На печать выдается переход к новой строке
ECHO	X ECHO	Символ с кодом X выдается на печать и удаляется из стека
HEX, DEC, OCTAL		Переключение систем счисления для оператора печати
Операторы цикла и ветвления		
DO	X У DO	Оператор начала цикла: в аппаратном стеке формируются три параметра цикла: адрес возврата, количество повторений, равное X—У, начальное значение управляющей переменной, равное У X и У из стека (операндов) удаляются
LOOP	LOOP	Проверка окончания цикла: если цикл не окончен, значение управляющей переменной наращивается на единицу и управление передается на начало цикла По окончании цикла (значение управляющей переменной равно или больше параметра X оператора DO) из аппаратного стека удаляются параметры цикла (три элемента), выполняется следующий оператор
PLUSLOOP	X PLUSLOOP	Оператор, аналогичный LOOP, но значение управляющей переменной наращивается на X
I	I	Текущее значение управляющей переменной помещается в стек
IFZ, IFNZ IFC, IFNC IFP, IFM		Ветвление вперед по признакам нуля, переноса, знака, четности до оператора ENDIF или ELSE соответствующего уровня
IFPE, IFP		Программный код между операторами IF... и ENDIF или IF... и ELSE выполнить, если условие ветвления выполняется
ENDIF	ENDIF	Конец действия операторов IF... (невыполняемый оператор)

Окончание табл. 1

Оператор	Форма записи	Выполнение
ELSE	ELSE	Программный код между ELSE и ENDIF выполняется при невыполнении условия ветвления в операторе IF... (выполняемый оператор)
BEGIN	BEGIN	Обозначает точку ветвления назад (невыполняемый оператор)
WZ, WNZ WC, WNC WP, WM WPE, WPO		Условное ветвление назад до оператора BEGIN соответствующего уровня

Базовый набор операторов

В табл. 1 приводится состав базового набора операторов, за исключением команд ассемблера, которые могут быть различными в зависимости от используемой микроЭВМ. Пользователь может не включать в рабочую программу все (или часть) операторы базового набора, а освободившиеся имена и память использовать по своему усмотрению. Аргументы операторов, предварительно помещаемые в стек, обозначены буквами X и Y, причем аргумент X находится на верхушке стека. Оба аргумента — 16-битовые целые числа.

Имена операторов имеют латинское написание в связи с тем, что, во-первых, это общепринятые обозначения, как и для ассемблерных команд, во-вторых, многие операторы по имени и выполняемой функции соответствуют операторам языка Форт. Напротив, операторам, введенным пользователем, рекомендуется присваивать имена, написанные кириллицей, чтобы яснее отразить выполняемую функцию.

Обратная польская запись применяется в языке Диалог при использовании только операторов базового набора и операторов, введенных пользователем, команды ассемблера записываются в привычной инфиксной записи. В языке Форт команды ассемблера применяются в польской записи. Это создает неудобства при изучении языка, к тому же применение этих команд разрешено только в операторах CODE. В языке же Диалог команды ассемблера используются в любом месте программы.

Наличие интерпретатора в системе Форт неизбежно замедляет выполнение программ, особенно при реализации на микропроцессоре К580ИК80, не имеющем в системе команд косвенного автоинкрементного способа вызова подпрограмм (как, например, в микроЭВМ «Электроника 60»). В системе Диалог отсутствует какой-либо интерпретатор, поэтому программы выполняются всегда в реальном времени.

В системе не существует ограничений на длину имен — распознавание ведется по всем символам, составляющим имя: таблица имен, существующая во время отладки, отбрасывается во время исполнения и не занимает память. Запись констант в языке Диалог однозначна (например, «159» всегда обозначает десятичную константу). В Форте же необходимо просматривать листинг, чтобы выяснить, не было ли переключения на другую систему счисления (HEX, OCTAL).

Создание рабочих программ

Создание ПО можно условно разделить на три этапа,

Первый этап:

написание части программы в виде комплекта новых операторов (подпрограмм);

ввод с терминала первого оператора;

пробный запуск оператора и поиск ошибок;

коррекция исходного текста, находящегося в буфере, с помощью встроенного редактора текстов (при выходе из редактора автоматически запускается транслятор);

пробный запуск и коррекция исходного текста необходимое число раз;

вывод отлаженного исходного текста на внешний носитель и распечатка на АЦПУ;

ввод исходного текста следующего оператора, отладка, коррекция и вывод текста.

Результат первого этапа — это полностью или частично отлаженный текст.

Второй этап:

написание новых операторов;

ввод (и трансляция) ранее определенных операторов (трансляция однопроходная);

при необходимости коррекции ранее определенного оператора исходный текст его вводится, редактируется и транслируется в машинные коды. После пробного запуска любой части программы при правильной работе исходный текст выводится на внешний носитель;

при получении достаточно большого количества отлаженных операторов специальной командой формируется отладочный модуль. В него входит система Диалог, дополненная операторами пользователя. Язык, таким образом, расширяется.

Третий этап:

в память ЭВМ загружается отладочный модуль (расширенный язык);

вводятся и отлаживаются новые операторы (или заменяются старые), входящие в отладочный модуль. Один из операторов назначается головным, вызывающим остальные операторы;

окончив комплексную отладку, пользователь сообщает системе имя головного оператора и по специальной команде выводит загрузочный файл всей программы с автозапуском. Программа пригодна для записи в ПЗУ.

Таблица 2

Операторы управления компиляцией и отладкой программ

LOAD	Загрузить с внешнего носителя исходный текст, компилировать его и разместить в памяти. Загрузка прекращается после ввода оператора END
LDOP	Оператор LOAD для одного определения
LIST	Распечатать последнее определение
TEXT	Вывести на внешний носитель последнее определение
EDIT	Редактировать последнее определение
NEW	Стереть программу и таблицу имен
EOF	Закрыть файл вывода исходного текста
EXIT	Выйти в операционную систему
OPERS	Распечатать адреса и имена операторов
VARS	Распечатать адреса и имена переменных, массивов и констант
MEMORY	Распечатать распределение памяти
FLUSH	Генерировать систему Диалог, дополненную операторами пользователя
WRITE	Генерировать окончательный вариант рабочей программы

Заключение

Четырехлетняя эксплуатация системы Диалог выявила ее достоинства при совместной отладке ПО и аппаратуры, управляемой встроенным МП. Особенность такого режима отладки в том, что часто необходимо многократно (сотни раз) изменять участок программы и запускать его для устранения неполадок в аппаратуре. Другое важное свойство языка — его близость к ассемблеру. Это упрощает изучение языка после ознакомления с ассемблером и позволяет реализовать практически все приемы программирования, разработанные для ассемблера, а «узкие» места в программе писать только в машинных кодах.

Система программирования Диалог записана на перфоленте в объектных кодах и адаптирована для отладочного комплекса «СО-01». Предполагается адаптировать ее для микроЭВМ «СМ-1800» (требуется перераспределение памяти).

Разрабатывается вторая версия системы: с инфиксной записью операторов, лучшей читаемостью программ, со средствами работы с массивами, с числами удвоенной точности. Вторая версия должна адаптироваться под любую операционную систему. В то же время программы должны работать под управлением простейшей перфоленточной операционной системы или без нее. Система Диалог второй версии разрабатывается на базе языка Диалог первой версии.

На языке Диалог реализовано ПО (228К байт) оверлейной структуры для автоматизированного комплекса измерений.

Управляющая программа комплекса — это специализированная операционная система (около 8К байт).

Кроме того, на языке Диалог написан с использованием только машинных команд редактор текстов. По сравнению с распространенным для микроЭВМ редактором текстов он занимает меньший объем памяти (менее 3К байт), при групповом редактировании работает в 10 раз быстрее, имеет дополнительный сервис и т. д. Итак, на языке Диалог возможна разработка малых и больших программ, включая и системные

Для справок о приобретении документации обращайтесь по домашнему телефону автора: 35-36-51 (г. Запорожье).

ЛИТЕРАТУРА

1. Хикс С. М. Лаконичность программ — основное достоинство языка Форт. — Электроника, 1979, № 6, с. 44—50.
2. Фаулджер Р. Программирование встроенных микропроцессоров. — М.: Мир, 1985.
3. Хротко Г. Языки программирования высокого уровня для микроЭВМ. — М.: Международный центр научной и технической информации, 1985.

Статья поступила 3 ноября 1985 г.

РЖ АВТ-85

6Б544. **Память с питанием от батарей для микропроцессора К1520.** — Radio — Fernsehenelektron., 1984, 33, № 12, 766—768 (нем.).

Сообщается об основных направлениях проектирования накопителей с ОЗУ на КМОП-элементах с питанием от батарей. Представлено конкретное решение для МП типа К1520, который содержит ОЗУ с максимальной емкостью 8К байт и с макс. емк. 32К байт и 40К байт. Приведено описание временных параметров, структур и выполняемых функций. Практическое опробование данных принципов было проведено в МП-системе типа МС-80.

АВТОМАТИЧЕСКАЯ КЛАССИФИКАЦИЯ ПАРТИЙ

(К ст. Дудоладова А. К.)

Система обработки шахматной информации «Дебют» ориентирована на пользователей-шахматистов различной квалификации. Она предназначена для информационной поддержки теоретической подготовки шахматистов к соревнованиям, начального обучения теории дебютов, демонстрации позиций и партий при анализе и обучении, подготовки шахматной информации к выходу в печать.

Состав системы: блок ввода, демонстрации и анализа партий; подсистема поиска партий в базе данных по различным признакам; подсистема обучения дебютам; программа автоматической классификации партий по дебютам; подсистема обработки результатов турниров.

База данных системы включает информацию о партиях (фамилии шахматистов, наименование соревнования, результат партии и др.), их тексты и классификатор, в соответствии с которым происходит индексация.

Основные функции системы:

поиск партий в базе данных по произвольной совокупности признаков (фамилиям шахматистов, индексу, результату, соревнованию, году, стране). Поиск партий, содержащих указанную позицию;

— запоминание и поиск позиций из партий, этюдов и задач;

— средства цветного графического отображения шахматных диаграмм на экране дисплея. Автоматическое разыгрывание партий и вариантов;

— средства анализа отложенных позиций (запоминание рассмотренных вариантов, их повторение, выдача на печать и др.). Использование манипулятора «мышь» для перемещения фигур на экране;

— автоматическая классификация партий по дебютам (система индексации Рабара);

— поддержка одновременной работы с двумя партиями и позициями позволяет проводить сравнительный анализ партий и позиций, рассматривать комментарии к партиям;

— обучение дебютам в пределах классификатора Рабара;

— расчет рейтингов и обработка турнирных таблиц;

— предусмотрен режим игры двух соперников с автоматической записью текста партии и отсчетом времени;

— тексты партий, позиции на экране, разделы классификатора могут быть выведены на печать.

Система разработана в Вычислительном центре АН СССР для персональных компьютеров, совместимых с IBM-PC, в частности функционирует на ПЭВМ ЕС-1840 с графическим дисплеем.



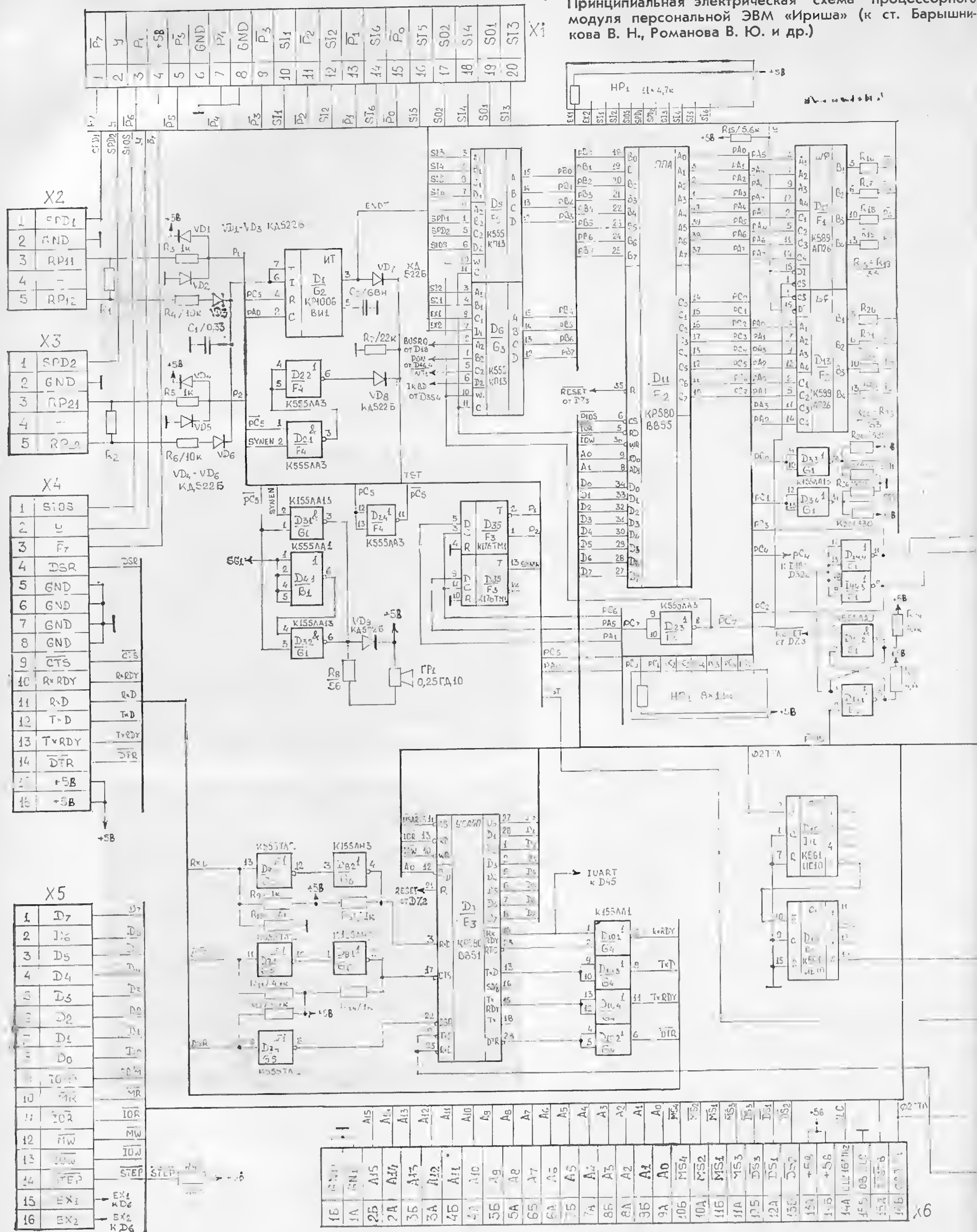




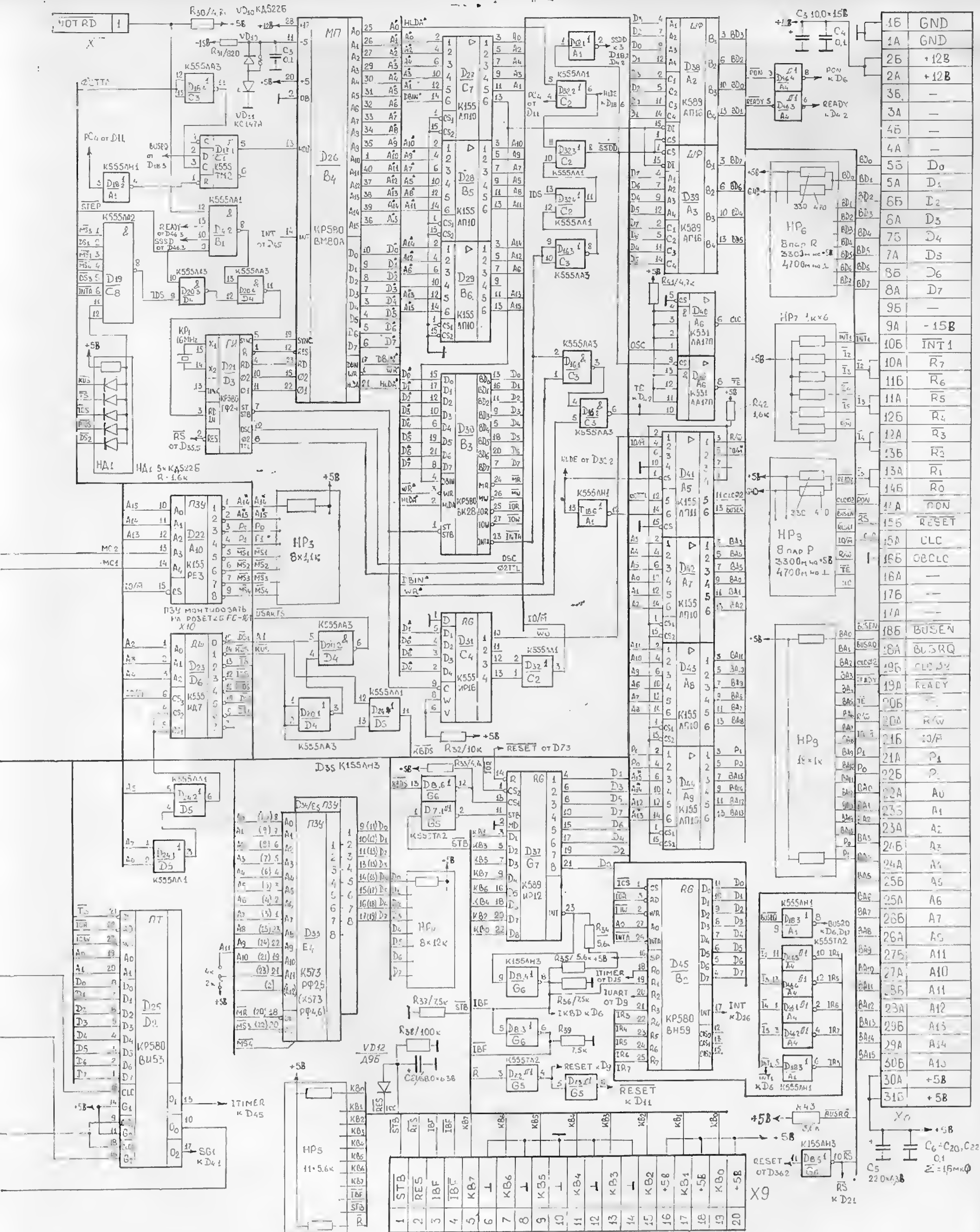
Расстановка позиции. Пользуясь манипулятором «мышь», с помощью стрелочного курсора можно выбрать фигуру и указать ее положение на доске

В тексте партии выделен дебютный вариант, идентифицированный по классификатору Рабара

Указание цветом пешечной структуры позиции при обучении



Принципиальная электрическая схема процессорного модуля персональной ЭВМ «Ириша» (к ст. Барышникова В. Н., Романова В. Ю. и др.)





КОМПЬЮТЕРНАЯ ЖИВОПИСЬ

(К ст. Курдюмова Г. Л.)

В отличие от известного устройства Mouse (мышь), трехмерная ручка управления — назовем ее Bird (птица) — дает оператору большие возможности для выражения эмоциональных состояний и развития художественного мастерства в компьютерной графике. Для введения в память ЭВМ информации о распределении цветовых оттенков в плоскости «картины» можно предоставить оператору большой выбор «инструментов», в том числе, иммитирующих традиционные инструменты художников. Перемещая в пространстве ручку управления, обозначающую, например, кисть, можно легко менять ширину оставляемого ею «следа», в зависимости от расстояния от конца ручки до воображаемой плоскости картины. При движении «пульверизатора», меняются как радиус «области напыления», так и его скорость. Моделирование свойств традиционных инструментов изобразительного искусства призвано обеспечить приемственность навыков работы и расширить круг пользователей, хотя «инструменты» могут быть иными, не имеющими механических аналогов. Можно, например, с помощью двух «птиц» в двух руках оператора одновременно вести линию на картине и точно управлять цветом вновь наносимых ее участков. Такого же эффекта можно достичь с одной ручкой, реагирующей на 6 степеней свободы ее перемещения: пусть, например, положение конца ручки определяет место и размеры оставляемого в данный момент цветового пятна, а углы ее поворота — цвет «краски». Участки картины, требующие детальной проработки, можно «растягивать» (то есть менять масштаб отображения), выписывать в них мелкие детали, а затем — снова сжимать. Это позволяет писать картины более тонкие, чем способен одновременно показать экран дисплея. Их можно выдавать на печать фрагментами, используя последующее совмещение. Подобные методы могут найти применение, например, в архитектуре.

УДК 681.3—181.1:06

В. В. Куклин, Н. М. Калинин, Ю. А. Бобров

ОТЛАДОЧНЫЙ МОДУЛЬ НА БАЗЕ ОДНОКРИСТАЛЬНОГО МИКРОПРОЦЕССОРА К1801ВМ1

Отладочный модуль имеет небольшие габариты, малую потребляемую мощность и программу монитор, записанную в УФРПЗУ. Модуль предназначен для первоначального ознакомления с БИС микропроцессора К1801ВМ1, его архитектурой и системой команд. Кроме того, при увеличении нагрузочной способности интерфейса его можно подключать к магистрали отлаживаемых микропроцессорных систем.

Отладочный модуль состоит из микроЭВМ на базе ОМП К1801ВМ1, индикационно-клавишного устройства «Электроника С5-2107» и блока питания напряжением 5 В (рис. 1). МикроЭВМ размещена в корпусе из плексиглаза размерами 170×145×40 (мм) (рис. 2). Потребляемая мощность не превышает 3 Вт.

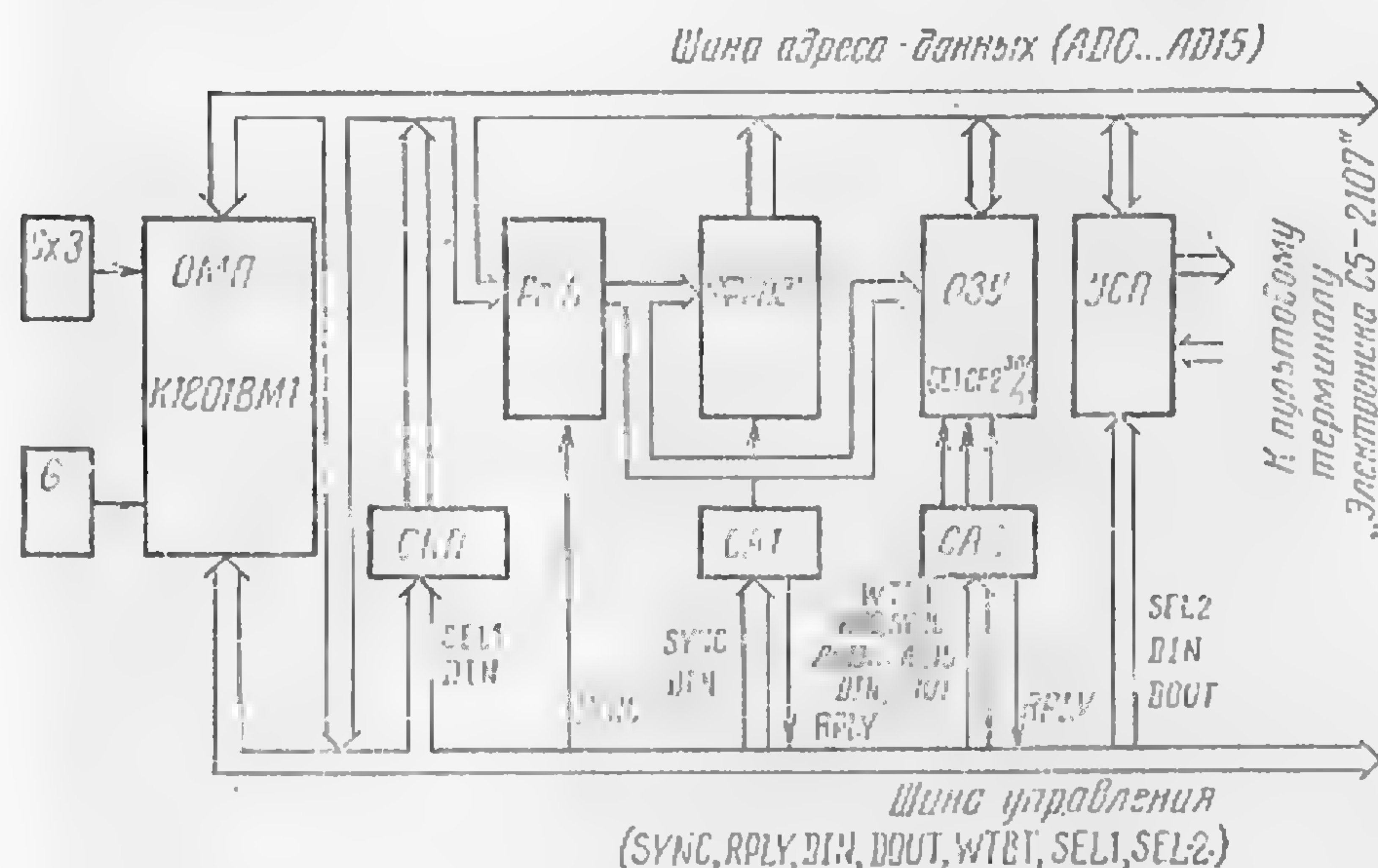


Рис. 1 Структурная схема отладочного модуля:
Сх. 3 — схема запуска; СНП — схема начального пуска; СА1, СА2 — селекторы адреса; УСП — устройство согласования с пультом

Рис 2. Внешний вид отладочного модуля

Принципиальная схема отладочного модуля (рис 3). В разработанном отладочном модуле применены интерфейсные схемы с малыми входными токами, многоразрядная память и микросхемы К588ВГ2 в качестве селекторов адреса.

Однокристальный микропроцессор (D1) управляет работой микроЭВМ, используя сигналы собственной магистрали [1]. Интерфейсные схемы выполнены на ИС К555, так как ток нагрузки на выходе D1 не должен превышать 3,2 мА. Для обеспечения высокого логического уровня на магистрали устанавливаются резисторы R11...R33. Частота синхронизации равна 2,5 МГц.

Схема начального пуска (D3.2, D3.3, D3.4) запускает D1 в момент включения питания с помощью цепочки R4, C3 или по внешнему сигналу Пуск низкого уровня. Цепочка R5, C4 служит для сдвига сигнала AI CO относительно DLCO. Стартовый адрес D1 100000_8 достигается замыканием сигналов SEL1 и AD15.

Запоминание адреса в регистре адреса (D5, D6) производится по отрицательному перепаду сигнала SYNC через инвертор D3.5.

Селекторы адреса (D7, D8) формируют сигналы управления обменом информации между D1 и ОЗУ (D9...D12), D1 и УФРПЗУ (D13, D14). Так как под ОЗУ и УФРПЗУ выделены зоны по 4 К (ОЗУ начинается с адреса 000000_8, ПЗУ — 100000_8), то дешифрация может быть неполной.

Для минимизации аппаратных затрат УСП входной порт (D17) с тремя устойчивыми состояниями и выходной порт (D15, D16) имеют одинаковый адрес (177714_8), соответствующий регистру расширения ввода-вывода D1 (сигнал SEL 2). Разряды выходного порта (D15, D16) используются следующим образом: 0...7 — соответственно коды сегментов A, B, C, D, E, F, G, H семисегментного индикатора; 8...11 — коды номера зажигаемого индикатора. В состав УСП входят также схема селекции (D2.3, D2.4) и дешифратор номера индикатора (D18, D19).

Для отображения результатов выполнения операций используется имеющийся в пультовом терминале однострочный дисплей, содержащий 13 семисегментных индикаторов динамического типа (рис. 4). Код 00_8 вызывает зажигание первого знакоместа (разряда), а код 14_8 — тринадцатого. Введение дополнительных ключей (рис 5) позволяет отображать на индикаторе пульта до 13 знаков. Конструктивно ключи можно разместить на плате микроЭВМ или внутри пульта (рис. 6). Схемы преобразователей уровня модуля служат для согласования выходных сигналов микроЭВМ (уровня ТТЛ) с входными управляющими цепями индикаторного устройства.

Программа монитор. Монитор предназначен для подготовки, редактирования и контроля выполнения программ пользователя в машинных кодах (объем программы 1К). Достоинства его — широкие функцио-

Рис. 3. Принципиальная схема отладочного модуля

нальные возможности, наличие защиты от ошибочных действий оператора с выдачей диагностических сообщений о допущенных ошибках, а также возможность создания библиотеки подпрограмм, ориентированных на конкретные задачи пользователя.

Все поле индикации на пультовом терминале разделяется на поле адреса и поле данных, содержащие каждое по шесть знакомест, а также поле режима, состоящее из одного знакоместа и расположенное между полями адреса и данных. Поле режима предназначено для вывода информации о выполненной операции, а также о наличии ошибок при работе с отладочным модулем, причем индикация в этом поле происходит в режиме мерцания. Сегмент зажигается в знакоместе при записи «Лог. 1» в соответствующий разряд (0...7) адреса пульта. Монитор выдает сразу два байта, поэтому регистр адреса пульта может не иметь байтовой адресации.

14_8	13_8	12_8	11_8	10_8	07_8	06_8	05_8	04_8	03_8	02_8	01_8	00_8	Код N знакоместа
13	12	11	10	9	8	7	6	5	4	3	2	1	Номер знакоместа
Поле адреса						Поле режима	Поле данных						

Э 8 П С О U ? Б

Рис. 4. Формат поля индикации. Функции поля режима:

Э — запись; 8 — ввод точки останова, НА подсчета КС;
П — проверка точки останова, НА подсчета КС;
С — исключение точки останова; О — останов программы в точке останова; U — пошаговое выполнение программы;
? — ошибка; Б — работа с байтами

Рис. 5. Схема дополнительных ключей

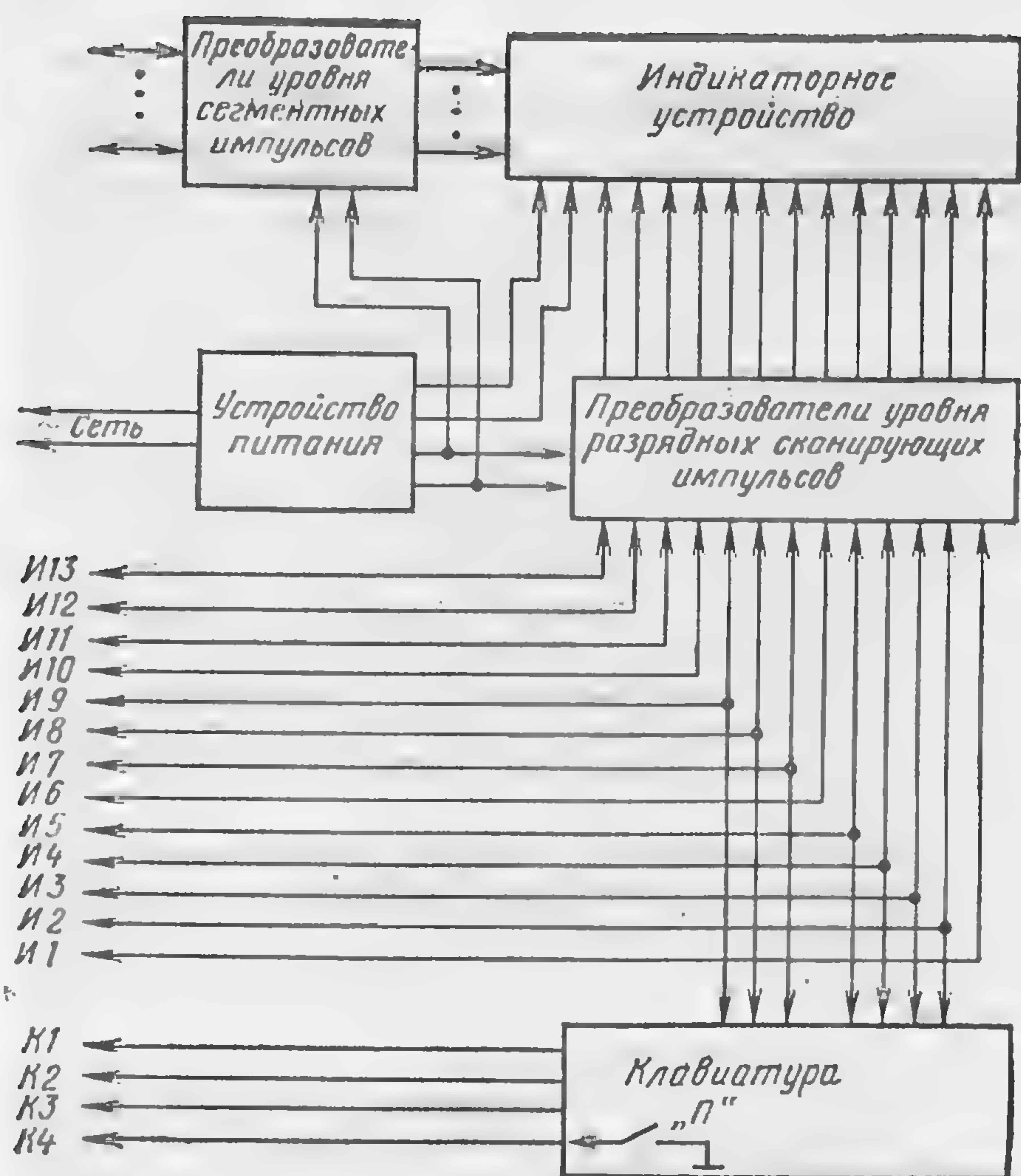


Рис. 6. Функциональная схема модуля пульта

Клавиатура пульта выполнена в виде матрицы 4×6 клавиш. Опрос клавиатуры производится одновременно с отображением определенных знакомест многоразрядного индикатора (рис. 7), при этом код строки Код1...Код4 считывается через входной порт пульта. Нажатой клавише соответствует сигнал низкого уровня (проверка их в строке происходит сверху вниз).

Для выполнения операций в мониторе используются восемь цифровых клавиш, клавиша Р для работы с регистрами общего назначения (РОН) и регистром состояния процессора (РСП) программ пользователя, клавиша Б для организации адресации по байтам и 13 командных клавиш. Ввод цифровых данных с клавиатуры, так же как и вывод на индикацию, производится в восьмеричной системе счисления. Цифровая информация, набираемая на клавиатуре, интерпретируется монитором либо как адрес, либо как данные в зависимости от предыдущей выполняемой операции, а также от нажимаемой впоследствии командной клавиши. Она запоминается соответственно в буферах адреса (БА) или данных (БД) и отображается в соответствующем поле индикации. После выполнения операций адрес операнда сохраняется в БА, поэтому нет необходимости повторно набирать его для выполнения других операций над этим операндом.

		01_8	02_8	03_8	04_8	06_8	07_8	10_8
Код1		0	1	2	3	Р		
Код2	П	4	5	6	7	ЗБ	Б	
Код3		ЧТП	ЧТ	ЧТС	ЧТК	ЧТО	ЗКР	
Код4		ОСТ	ОТМ	ПСК	ШАГ	НЛ	КС	

Рис. 7. Клавиатура пульта

После запуска монитора в младшем разряде поля данных горит 0 — режим готовности. При повторном перезапуске (клавиша П) информация о введенных точках останова и содержимом РОН программы пользователя сохраняется.

Функциональные возможности монитора (в скобках указаны командные клавиши пультового терминала, соответствующие данным операциям):

— чтение содержимого (открытие) заданной ячейки памяти или регистра ввода-вывода (ЧТ);

— чтение содержимого предыдущей и последующей ячеек памяти (регистров ввода-вывода) (ЧТП и ЧТС). При этом БА ← БА±2;

— чтение содержимого ячейки памяти с использованием косвенной адресации (ЧТК). Адрес операнда равен содержимому предыдущей открытой ячейки БА ← (БА);

— чтение содержимого ячейки памяти с использованием относительной адресации (ЧТО). Адрес операнда равен сумме содержимого открытой ячейки по адресу АДР и содержимого счетчика команд (СК), равного АДР+2, БА←(БА)+СК;

— изменение содержимого ячеек памяти. После чтения содержимого ячейки на цифровой клавиатуре набирается новое значение и нажимается клавиша (ЗКР); при этом происходит запись нового содержимого в ячейку (закрытие ячейки). Если нажимается одна из клавиш (ЧТП, ЧТС, ЧТК, ЧТО), то кроме закрытия ячейки выполняются действия, определяемые нажатой клавишей. Кроме того, клавиша ЗКР (при отсутствии перед ее нажатием ввода с цифровой клавиатуры) используется для перевода монитора в режим готовности;

— чтение и изменение содержимого ячеек памяти с адресацией по байтам (Б). Выполняются все предыдущие операции, за исключением ЧТК и ЧТО (в противном случае выдается сообщение об ошибке). При нажатии ЧТП или ЧТС БА ← БА ± 1. Режим адресации по байтам отменяется повторным нажатием клавиши Б;

— чтение и изменение содержимого РОН программ пользователя, РСП (ЧТ, ЧТП, ЧТС, ЗКР). Для обращения к РОН нажимается Р и номер регистра, к РСП — дважды нажимается Р (при этом в поле адреса отображается РС);

— исключение последней, ошибочно набранной на клавиатуре цифры (забой) (ЗБ);

— ввод точек останова в программу пользователя с автоматическим присвоением им номеров (0...7) (ОСТ). Адрес операнда запоминается в таблице точек останова, в поле данных индицируется номер точки останова;

— проверка введенных точек останова по их номерам. Вводится номер точек останова, нажимается ОСТ. В поле адреса индицируется адрес точки останова (или 0, если она не введена), в поле данных — ее номер. Повторное нажатие ОСТ вызывает проверку точки останова со следующим номером;

— исключение введенных точек останова как по адресам их расположения, так и по их номерам (ОТМ). Для исключения всех точек останова перед нажатием ОТМ набрать 10;

— выполнение программы пользователя с любого адреса (ПСК). При этом

ССП ← 0
СК ← БА,

где ССП — слово состояния процессора. Если последней нажатой перед ПСК клавишей была ШАГ, то выполняется

ССП ← РСП
СК ← БК,

где БК — буфер команды, содержащий адрес очередной выполняемой команды программы пользователя. Это позволяет выполнять программу с заданным значением ССП. При достижении точки останова индицируется ее адрес и номер, а также выполняется

РСП ← ССП
БА ← СК
БК ← СК

Перед выполнением данной операции, а также пошагового выполнения необходимо установить вершину стека пользователя (регистр Р6), адрес которой не может быть меньше 400_8;

— выполнение программы пользователя в режиме прослеживания (пошагового выполнения) (ШАГ). Если адрес операнда был набран непосредственно перед нажатием клавиши ШАГ, то

ССП ← РСП
СК ← БА

В остальных случаях

ССП ← РСП
СК ← БК

После этого происходит выполнение одной команды программы пользователя и индицируется адрес следующей команды и его содержимое. Кроме того, выполняется

РСП ← ССП
БА ← СК
БК ← СК

Введенные точки останова не влияют на выполнение команд в режиме ШАГ. Поэтому для продолжения выполнения программы пользователя после достижения точки останова в режиме ПСК достаточно последовательно нажать ШАГ и ПСК;

— ввод начального адреса подсчета контрольной суммы (НА). Выполняется

БНА ← БА,

где БНА — буфер начального адреса;

— проверка введенного начального адреса (НА). Данная операция выполняется, если монитор находился в режиме готовности;

— контрольное суммирование содержимого памяти, начиная с введенного начального адреса и кончая заданным включительно (КС). Начальный адрес перед нажатием КС должен находиться в БНА, конечный — в БА. Введенные точки останова не влияют на значение контрольной суммы;

— возможность выполнения перечисленных операций или их комбинаций в режиме многократного выполнения. Данный режим включается, если одна из клавиш остается нажатой более 2 с. Это позволяет производить многократный спрос регистра ввода-вывода, автоматическое обнуление области ОЗУ (если нажаты клавиши 0 и ЧТС) и др. Обработка нескольких нажатых клавиш производится циклически в порядке их опроса;

— защита от ошибочных действий оператора, сообщение об ошибках. Защита предусмотрена для повышения надежности работы монитора, а также для предохранения программы пользователя от возможных искажений. В этом случае монитор выдает диагностическое сообщение о допущенной ошибке, причем в поле данных индицируется код ошибки. Коды ошибок, распознаваемых монитором: 0 — несуществующий адрес операнда (ошибка обращения к каналу), 1 — несуществующий код команды (ошибка по резервной команде), 2 — запрещенная клавиша (нарушение логической последовательности нажатия клавиш), 3 — недопустимые действия при адресации по байтам, 4 — недопустимые действия при работе с РОН, 5 — недопустимый номер РОН, 6 — нечетный адрес операнда при отсутствии режима адресации по байтам, 7 — точки останова не введены, 10 — переполнение таблицы точек останова, 11 — повторный ввод точки останова по одному адресу, 12 — точка останова с указанным номером не введена, 13 — точка останова по указанному адресу не введена, 14 — неверно определена вершина стека пользователя;

— отображение результатов выполнения перечисленных операций на индикаторах пульта. Символы, отображаемые в поле режима, приведены на рис. 4.

Программу монитор можно разделить на ядро программы и драйвер пультового терминала, организующий связь программы с оператором. В ядре программы выполняются все действия по определению типа операции и операнда, их допустимости, анализу таблицы точек останова, выполнению операции и подготовке ОЗУ индикации.

Отладочные режимы в мониторе реализованы с помощью методов программного прерывания (в режиме пуска программы пользователя) и программно-командного прерывания (пошаговое выполнение программы) [2].

Метод программного прерывания основан на замене первого слова команды программы пользователя по адресу точки останова на команду прерывания TRAP с фиксированным адрес-вектором 34_8. Метод позволяет выполнять программы пользователя в реальном масштабе времени, для реализации его не требуется никаких дополнительных средств, кроме программных.

В методе программно-командного прерывания при установленном Т-бите ССП после выполнения каждой команды происходит прерывание с адрес-вектором 14_8. В мониторе перед выполнением очередной команды программы пользователя в режиме прослеживания устанавливается Т-бит ССП, после выполнения команды и возникновения прерывания он очищается и анализируется новое состояние программы пользователя.

Достоинство монитора — возможность создания библиотеки подпрограмм, размещаемой в свободной зоне ПЗУ и ориентированной на конкретные задачи пользователя, что существенно расширяет функциональные возможности отладочного модуля. В библиотеку могут входить подпрограммы обслуживания внешних устройств (например, ввода-вывода с перфоленты), подпрограммы преобразования информации, вычисления функций и т. д. Каждая подпрограмма должна заканчиваться командой TRAP. Необходимые параметры могут передаваться либо через РОН пользователя, либо через ОЗУ. Для выполнения подпрограммы достаточно выполнить пуск подпрограммы операции ПСК с адреса начала подпрограммы. После ее выполнения управление автоматически передается в монитор

Программа монитор реализована по принципу позиционно-независимого кодирования и является перемещаемой. Пусковой адрес монитора — адрес его загрузки (АЗ).

Для работы монитора необходимо ОЗУ обьемом 128 байт, причем предусмотрена возможность работы монитора с ОЗУ, расположенным в любой области адресного пространства. Адрес выбранной для ОЗУ монитора области памяти (АМ) записывается в ячейку с адресом (АЗ + 3772_8), при этом в мониторе используется зона памяти от (АМ—40_8) до (АМ + 140_8). Также имеется возможность изменения адреса пультового терминала. Для настройки монитора на конкретный адрес пульта последний необходимо записать в ячейку с адресом (АЗ + 3770_8).

В мониторе используются прерывания с адрес-векторами 4_8, 10_8, 14_8, 34_8, поэтому пользователю не рекомендуется применять их в своих программах

Распечатка программы монитор в машинных кодах:

```
DK:M.SAV/N/0:1
BLOCK NUMBER  00001
000/ 016700 003766 010005 012705 000004 010715 062725 002454
020/ 005025 010715 062725 002462 005025 010715 062725 001564
040/ 005015 012705 000034 010715 062725 001754 005015 026027
060/ 000026 123456 001411 012760 123456 000026 012760 000010
100/ 000030 012701 000032 000402 012701 000100 004767 002750
120/ 112760 000077 000135 012704 175100 000453 005704 100407
140/ 032704 020000 001016 022705 000007 002010 000402 005060
160/ 000116 004767 002676 012701 000126 000402 012701 000135
200/ 005003 022705 000007 002427 005760 000116 100011 030127
220/ 000135 001406 005703 001404 012705 000005 000167 001254
240/ 012704 060210 006303 006303 006303 050503 010305 004767
260/ 002636 020167 003104 022704 002010 001402 000167 000516
300/ 012760 100000 000115 012704 050300 020761 004767 030420
320/ 100403 012701 000130 002572 012704 160000 112760 010117
340/ 000127 [illegible] 000115 100201 000430 010205 004407 004767
360/ 002252 100440 001443 010205 012704 040000 012701 000136
400/ 004767 002514 012760 000110 100002 010205 000417 004767
420/ 002114 001402 010105 000401 010205 005760 000100 001424
440/ 111505 042705 177400 000403 004767 002360 011505 012701
460/ 000135 000675 005007 062360 000116 016005 000116 042705
500/ 177760 010502 006303 062700 000004 060002 062725 012705
520/ 000002 004767 002212 001414 010302 005760 000116 100010
540/ 042702 177760 060260 000116 006302 062701 000004 [illegible]
560/ 005760 000116 100407 005760 000100 001401 006205 060503
600/ 010205 [illegible] 060502 160003 022702 100002 103400 012702
620/ 000024 012760 100010 000116 000414 020227 200010 103406
640/ 012702 000004 012760 100000 000116 000403 006205 060500
660/ 000116 060002 016005 000116 042705 177760 000635 012705
700/ 177776 000707 005760 000100 001403 012705 000003 000441
720/ 032704 010000 001034 004767 002006 001404 005760 000115
740/ 100426 010302 005060 000116 032704 000004 001402 062702
760/ 000502 011202 000775 005760 000116 100003 012705 000004
```

```
DK:M.SAV/N/0:2
BLOCK NUMBER  00002
000/ 000410 052704 000004 000776 032704 000700 001004 012705
020/ 000002 000167 001466 006103 006103 006103 042703 160004
040/ 000167 177310 032701 001003 001101 032704 004000 001357
060/ 052704 000240 004767 001544 001741 103411 103472 004767
100/ 001732 004767 001430 001405 012705 000011 000742 000167
120/ 000442 005760 000030 001003 012765 000010 000732 016003
140/ 000032 004767 001650 010325 011325 012712 164400 005360
160/ 000030 005260 000032 005725 001405 005715 026027 000032
200/ 000007 103767 112760 000177 000127 012704 174000 010105
220/ 012701 000135 004767 001670 060001 112741 000271 112741
240/ 000077 012701 000120 012205 000167 177002 005203 042703
260/ 177770 022760 000010 000030 001003 012705 000007 000651
300/ 012704 172000 112760 000067 000127 004767 001500 011502
320/ 000736 112760 000071 000127 032704 000100 001230 022760
340/ 000010 000030 001752 052704 000040 004767 001256 001644
360/ 103471 102433 005702 001403 004767 001142 001003 012705
400/ 000013 000607 016003 000102 012704 170000 011112 005011
420/ 005041 005260 000030 016027 000030 000010 031410 026003
440/ 000032 002665 011360 000032 000062 012704 170000 012705
460/ 000034 060005 012701 000010 005725 001403 011555 005025
500/ 005015 005725 077107 012760 000010 000030 005060 000032
520/ 004767 001232 112760 000071 000127 012760 134477 000134
540/ 000157 001635 004767 001246 010501 005721 001403 016102
560/ 177776 000712 011705 000012 030704 052730 000000 000024
600/ 052704 000400 032704 000010 001033 016002 000076 000423
620/ 010677 003146 016706 002142 012746 170400 000501 032704
640/ 001000 001406 012705 000002 012760 000020 000424 000651
660/ 004767 000750 001441 004767 001142 032760 000001 000020
700/ 026027 000020 003402 107437 012765 000014 000755 016005
720/ 006120 052704 037400 [illegible] 016046 000014 032760 000020
740/ 000024 001403 004767 [illegible] [illegible] 005000 000002 000410
760/ 011112 010260 000002 000404 012765 000004 000725 005046
```

```
DK:M.SAV/N/0:3
BLOCK NUMBER  00003
000/ 010246 [illegible] 000020 014005 014004 014003 014002 014001
020/ 014000 000206 010077 [illegible] 016706 001736 012746 174000
040/ 010146 010046 016700 001722 012701 000004 060001 012671
060/ 012621 010221 010321 010421 012604 010521 011611 062721
100/ 000004 011603 [illegible] 012001 011311 005060 000116 032704
120/ 000403 001024 005742 010260 100076 112740 000077 000137
140/ 012705 000134 000205 005003 [illegible] 001003 000167 177376
160/ 005725 005207 020427 000010 103767 000416 042760 [illegible]
200/ 000024 005760 000002 001403 012770 104400 000002 010760
220/ 000076 112750 020076 000137 010305 000167 176136 032704
240/ 000100 001410 016002 000074 005260 000116 112760 000057
260/ 000127 000412 004767 000344 001677 004767 000576 [illegible]
300/ 000074 112760 000177 000137 013704 175000 004767 000044
320/ 012760 [illegible] 000134 004167 176710 004767 000275 [illegible]
340/ 004767 [illegible] [illegible] [illegible] 005004 010305 010002 [illegible]
360/ 000154 001403 061104 005723 000401 062304 030305 [illegible]
400/ 010503 [illegible] 012701 [illegible] 004767 000504 000001 [illegible]
420/ 000156 112741 [illegible] [illegible] [illegible] [illegible] [illegible] 176016
440/ 062760 [illegible] 000100 [illegible] [illegible] [illegible] 001403 [illegible]
460/ 000106 000137 [illegible] [illegible] [illegible] 016700 [illegible] [illegible]
500/ [illegible] [illegible] 000403 [illegible] [illegible] 004767 004767 [illegible]
520/ 012704 [illegible] [illegible] 000332 [illegible] 000167 175520 010446
540/ 026027 [illegible] 000010 001434 011701 000034 000001 005060
560/ 000102 010204 162104 001417 020437 000001 001002 005301
600/ 000412 005721 005000 [illegible] [illegible] 000137 000010 [illegible]
620/ 012504 [illegible] 000407 [illegible] [illegible] [illegible] 032704 [illegible]
640/ 001005 005760 000110 [illegible] [illegible] 000431 005760 000115
660/ 100005 020127 [illegible] 001423 [illegible] 000116 032704 000040
700/ 001404 020327 000010 002407 001410 010302 000242 000341
720/ 000244 000250 000404 000361 000773 000262 000270 000307
740/ 032704 [illegible] 001423 [illegible] 000135 [illegible] 010101 005740
760/ 000116 103411 004767 177546 001001 110201 005760 000100
```

```
DK:M.SAV/N/0:4
BLOCK NUMBER  00004
000/ 001402 110311 000401 010311 000264 000270 000207 010305
020/ 006305 006305 062705 000034 060005 000207 032702 000001
040/ 001405 012705 000005 010716 062716 176600 000207 012701
060/ 000120 000402 013701 000130 060001 105021 160001 032127
100/ 000135 101777 005301 000207 003277 047533 166546 003575
120/ 010246 010146 060001 010102 162702 000005 105022 030201
140/ 103775 005760 000116 100421 105021 010502 042702 177770
160/ 062702 062702 177726 111241 006005 006005 006005 042705
200/ 160000 001363 012601 012602 000207 022716 000126 103754
220/ 020527 000010 002422 122711 000163 001404 022760 100010
240/ 000116 001010 012703 000010 112761 000163 177777 112711
260/ 000071 000750 112711 000163 000745 112761 000163 177777
300/ 000723 000000 000004 001676 002044 000001 000005 001756
320/ 002322 000002 000006 001516 002636 000003 000007 001724
340/ 002572 004422 004422 004422 004422 000010 002010 001760
360/ 003236 004422 003440 001314 003332 010346 010146 010046
400/ 005760 000100 001406 105760 000127 001003 112760 000175
420/ 000127 004767 000250 042704 000020 012703 000400 005005
440/ 012702 000004 010377 000320 017701 000314 006001 103401
460/ 000425 020527 000033 001407 005205 020502 002767 022222
500/ 062703 000400 000757 032704 000020 001002 005060 000104
520/ 042704 040000 012760 177777 000110 000733 005760 000104
540/ 100407 032704 040000 001423 062760 000700 000104 000721
560/ 052704 000020 020560 000110 003734 012760 000003 000105
600/ 004767 000072 005360 000106 100373 010560 000110 052704
620/ 040000 012602 012601 012603 116000 000127 000106 105060
640/ 000127 006305 060705 062705 177434 011505 022705 000010
660/ 002403 000167 174246 060705 162705 004670 000115 012701
700/ 000135 060001 005060 000114 111160 000114 016077 000114
720/ 000046 012702 000064 077201 005301 105260 000115 132700
740/ 000006 000115 001004 062760 006000 000112 100365 126027
760/ 000115 000015 002731 003207 000000 000000 000000 000000
```

Монитор может быть использован также в отладочных модулях, реализованных на базе К588, «Электроника С5-41» и других микроЭВМ, совместимых по системе команд с рядом микроЭВМ «Электроника-60».

ЛИТЕРАТУРА

1. Дшхунян В. Л., Борщенко Ю. И., Науменков В. Р. и др. Однокристальные микропроцессоры комплекта БИС серии К1801. — Микропроцессорные средства и системы, 1984, № 4, с. 12—18.
2. Зеленко Г. В., Иванников А. Д., Сыпчук П. П. Проектирование и отладка микропроцессорных систем. — М.: Машиностроение, 1982.

Статья поступила 8 августа 1985 г.

УДК 681.326—181.4

В. М. Фельдман

ПРОВЕРКА И РЕМОНТ МИКРОЭВМ В УСЛОВИЯХ ЭКСПЛУАТАЦИИ

Учитывая повышенную сложность микропроцессорной техники и модульный принцип ее построения, предлагается организовать ремонт этой техники в два этапа [1]. На первом этапе неисправность выявляется с точностью до сменного модуля (типового элемента замены — ТЭЗ), а на втором ТЭЗ полностью ремонтируются с заменой составляющих их отказавших элементов. Второй этап, значительно более трудоемкий, требует квалифицированного персонала и специального дорогостоящего оборудования. Как правило, он централизован и охватывает определенный класс машин либо строится по отраслевому принципу.

Если главное условие эффективного ремонта на втором этапе — это глубина поиска дефекта, то на первом этапе необходимо преж-

де всего увеличить скорость восстановления работоспособности системы управления. При этом в большинстве случаев необходим ремонт на первом этапе без привлечения специалистов по вычислительной технике. Именно эти задачи первого этапа определили разработку приводимой ниже методики.

Известны [2] базовые способы обнаружения неисправностей:

метод разделения микроЭВМ на несколько частей, взаимно диагностирующих друг друга с помощью аппаратных средств и программного тестирования;

метод «наращивания правильности» или «расширяющегося теста» в сочетании с методом сравнения с «эталоном».

Суть первого способа ясна из наименования. Метод «наращивания правильности» предполагает поэтапное последовательное расширение работоспособной части системы путем включения в эту часть устройств, проверенных на предыдущем этапе. В ряде случаев, например при неисправности центрального процессора (ЦП), необходимо включать в систему заведомо исправные («эталонные») устройства.

Эксплуатационный персонал должен знать наименования ТЭЗ для микроЭВМ данной системы и их функциональное назначение, уметь запускать тесты и вести необходимый при этом диалог с системой, т. е. специальной подготовки по вычислительной технике не требуется.

Стенд проверки ТЭЗ строится на базе микроЭВМ, однотипной с проверяемой, и оснащается пультовым терминалом (ПТ), устройством ввода тестов (УВТ), интерфейсами с перечисленными устройствами и всеми ТЭЗ, входящими в состав микроЭВМ данной системы. Эти ТЭЗ — заведомо исправные («эталонные»).

Стенд подключается к проверяемой микроЭВМ платой-переходником с кабелем.

Чтобы организовать передвижной стенд для поиска неисправности непосредственно на месте установки системы, в качестве носителя тестового программного обеспечения рекомендуется использовать плату ПЗУ [3] (этим исключается необходимость в устройстве ввода с перфолент). Набор тестов зависит от конкретной конфигурации микроЭВМ.

Для измерений номиналов питающих напряжений, управляющих сигналов от вторичного источника питания (ВИП) и целостности монтажных соединений стенд оснащается комбинированным прибором.

По данной методике с помощью стенда неисправности находят с точностью до ТЭЗ, блока ВИП или каркаса микроЭВМ.

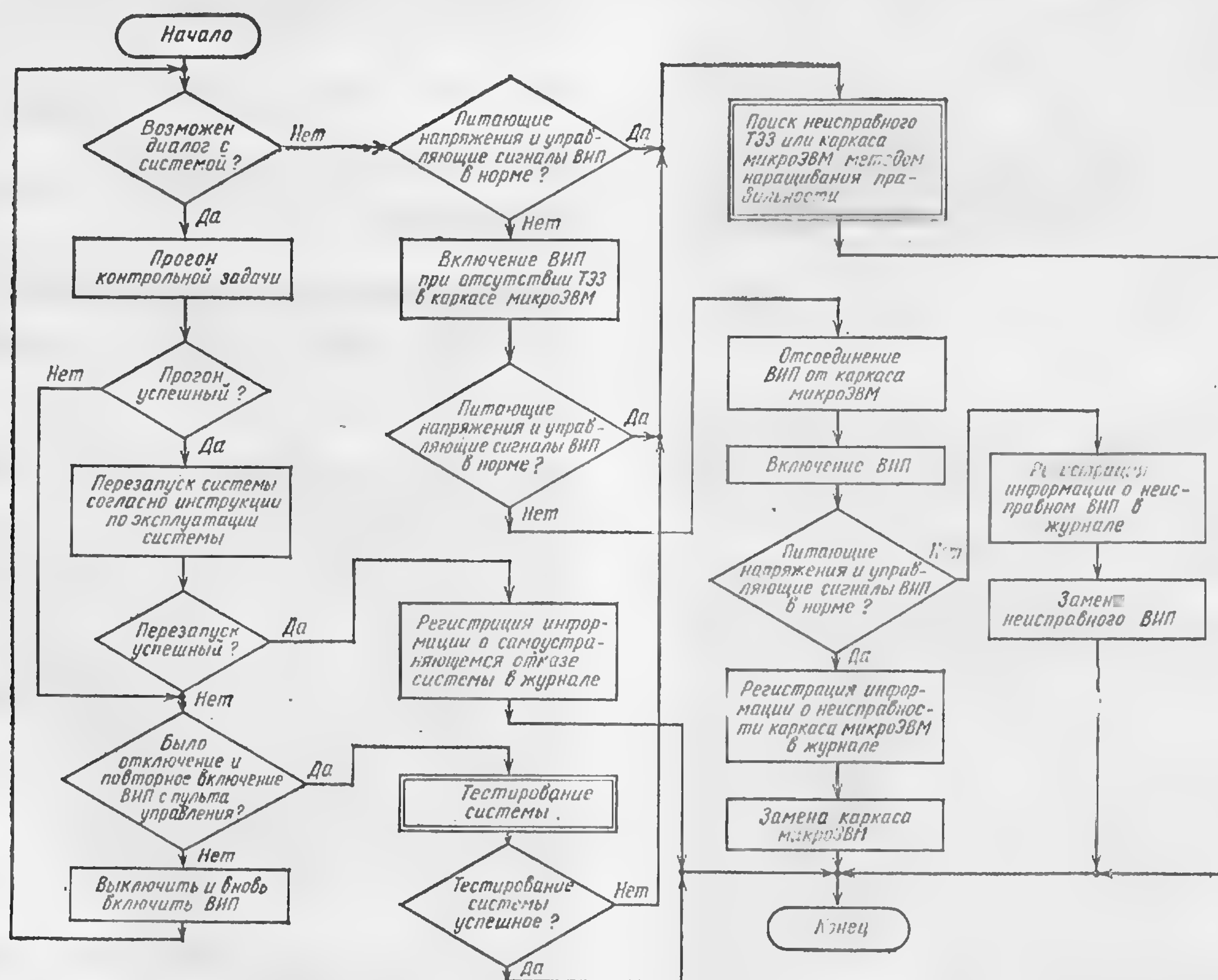


Рис. 1. Блок-схема алгоритма поиска неисправности с точностью до ТЭЗ, блока ВИП или каркаса микроЭВМ

В контуре управления микроЭВМ неисправности индицируются гашением индикатора «Питание» на передней панели управления микроЭВМ; аварийным переходом микроЭВМ в пультовый режим; «зависанием» системы (нет выдачи ожидаемой информации); выдачей информации о неисправности, обнаруженной системой диагностики.

В каждом из этих случаев необходимо отключить микроЭВМ от объекта управления (согласно соответствующей инструкции по эксплуатации системы) и найти неисправность (рис. 1).

Для удобства восприятия блок-схемы алгоритма отдельные блоки (обозначены двойными линиями) раскрыты в виде самостоятельных блок-схем алгоритмов.

Информацию об обнаружении неисправностей необходимо занести в «Журнал учета неисправностей при эксплуатации» (см. в формуляре на микроЭВМ).

Перед тестированием системы или поиском неисправности с помощью метода «наращивания правильности» необходимо визуально проконтролировать каждый проверяемый ТЭЗ и каркас микроЭВМ, чтобы выявить, например, сгоревшие элементы, нарушение пайки, контактную неисправность. Затем для каждого проверяемого ТЭЗ с помощью комбинированного прибора измеряются сопротивления между каждым контактом, выходящим на канальный разъем, и шинами питания, включая общую шину. При обнаружении короткого замыкания данный ТЭЗ отбраковывается и либо сразу ремонтируется, либо заменяется на исправный («эталонный»).

При проверке ВИП необходимо руководствоваться требованиями к питающим напряжениям и уровням управляющих сигналов, характеризующих состояние ВИП (они, как и схема подключения ВИП к ЭВМ, даны в инструкции по эксплуатации).

При тестировании системы (рис. 2) тесты вводят в память ЭВМ. Если в результате тестирования (в соответствии с эксплуатационной документацией на ЭВМ) однозначно выявляются неисправные ТЭЗ, то они заменяются исправными.

В остальных случаях для восстановления работоспособности системы пользуются методом «наращивания правильности» (рис. 3), выявлением и заменой.

После визуального контроля и поиска коротких замыканий восстанавливается работоспособность микроЭВМ в минимальной конфигурации (минимум устройств, необходимых для тестирования микроЭВМ). Если данная система управления включает в себя устройства, необходимые для минимальной конфигурации, то они должны быть задействованы в качестве испытуемых. В противном случае необходимо минимальную конфигурацию собрать с помощью стендового оборудования.

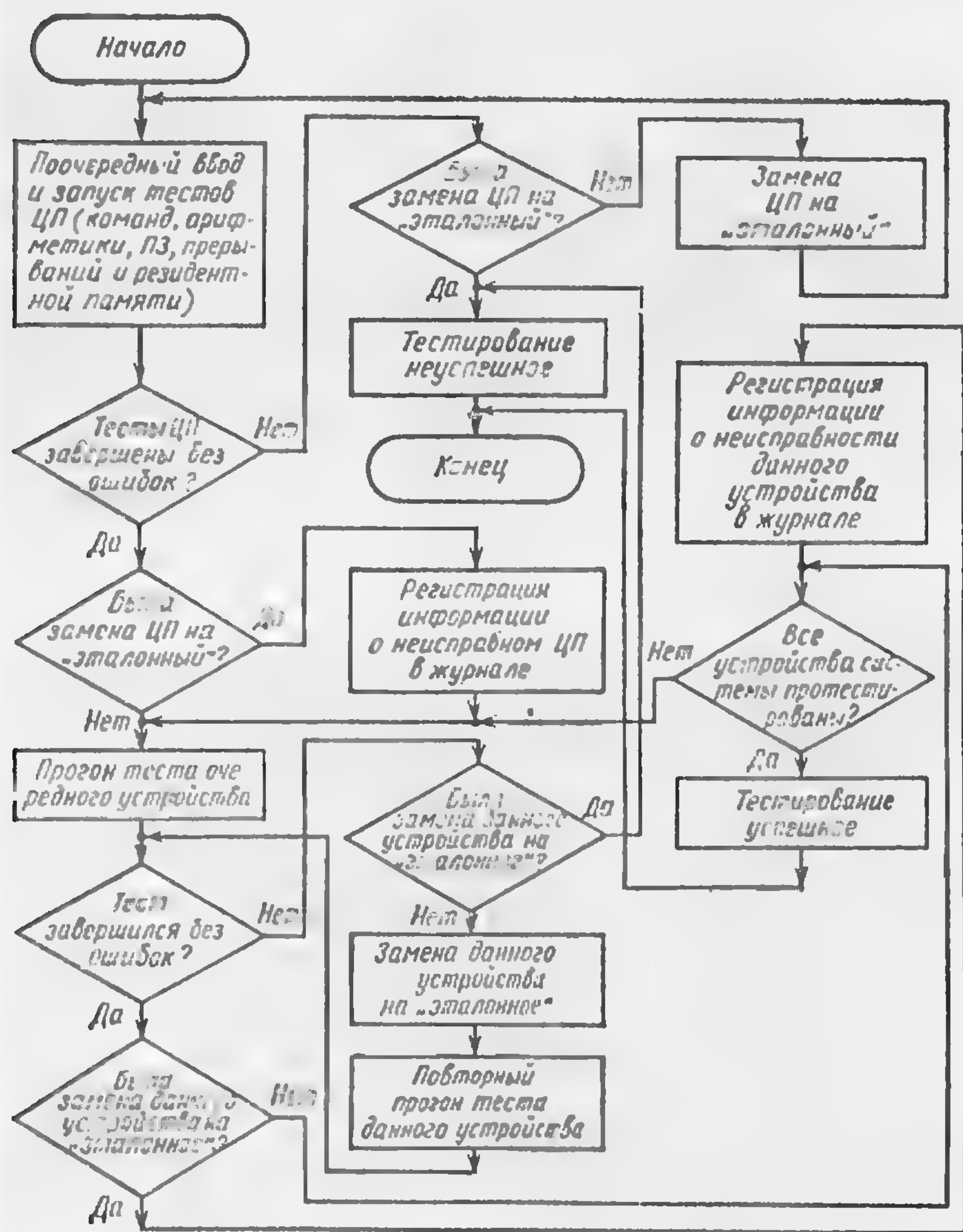


Рис. 2. Блок-схема алгоритма поиска неисправного ТЭЗ с помощью тестирования

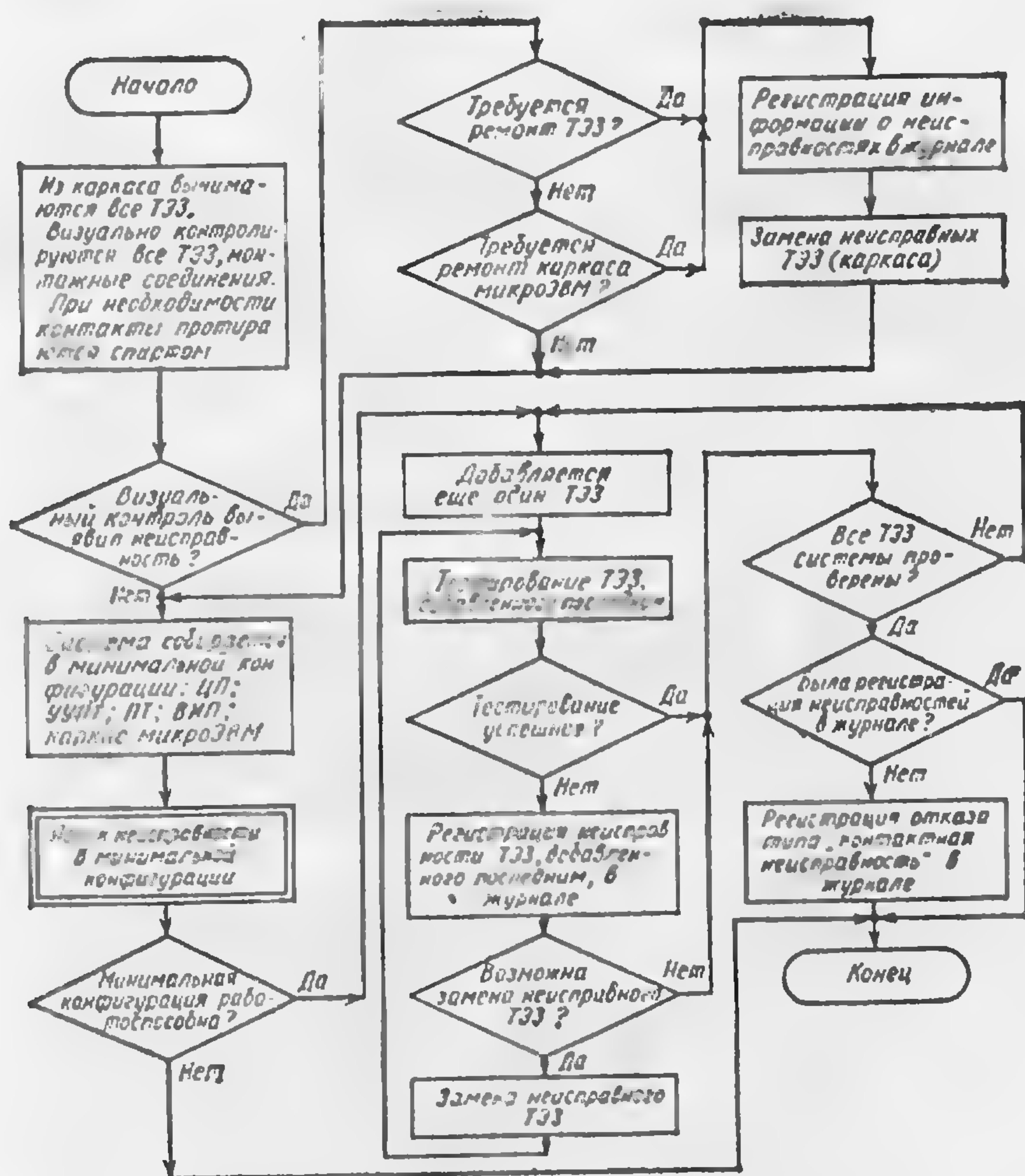


Рис. 3. Блок-схема алгоритма поиска неисправного ТЭЗ (или каркаса) микроЭВМ методом «наращивания правильности» работы

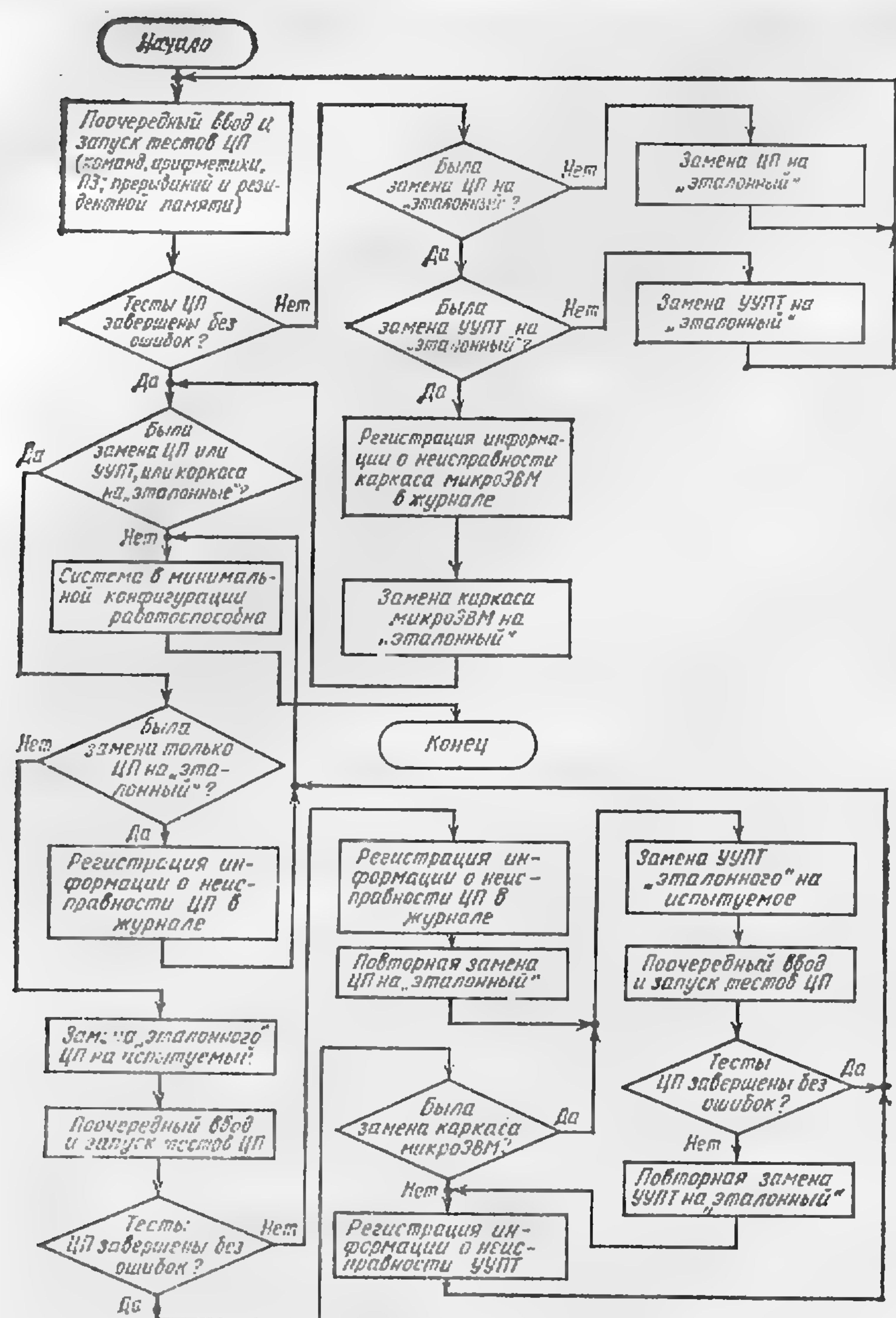


Рис. 4. Блок-схема алгоритма поиска неисправности в минимальной конфигурации

В алгоритме поиска неисправностей в минимальной конфигурации (рис. 4) было принято, что в состав системы входит микроЭВМ «Электроника 60», оснащенная ЦП, устройством управления пультовым терминалом (УУПТ), ВИП и рядом ТЭЗ, каркасом микроЭВМ, ПТ и УВТ.

Поиск неисправностей в минимальной конфигурации преследует две цели: однозначно определить неисправные ТЭЗ (или каркас микроЭВМ); собрать из исправных устройств работоспособную систему.

Достигнув работоспособности микроЭВМ в минимальной конфигурации, начинаем постепенно добавлять по одному испытуемому устройству (ТЭЗ), выявляя неисправные и заменяя их «эталонными», до сбора системы в требуемой конфигурации.

В некоторых случаях неисправности, возникшие в одних блоках, проявляются лишь при тестировании других. Например, возможна неисправность в блоке прерываний центрального процессора, которая проявится только при тестировании с одним из добавленных ТЭЗ. В таком случае даже добавление «эталонных» ТЭЗ не приведет к успешному тестированию. Необходима замена ЦП на «эталонный», а если это не изменит ситуацию — проверка испытуемых ТЭЗ в составе «эталонного» стендового оборудования. Когда тестирование испытуемого ТЭЗ успешно на стенде и неуспешно в составе испытуемой системы, необходима дополнительная проверка. С этой целью испытуемый ТЭЗ устанавливается на стенде, а «эталонные» поочередно заменяются на ТЭЗ из проверяемой системы, пока не выявится неисправный ТЭЗ. Если все ТЭЗ испытуемой системы проверены на стенде и тестирование успешное, надо заменить каркас проверяемой микроЭВМ.

Эксплуатация систем на базе микроЭВМ «Электроника 60» показала, что в подавляющем большинстве случаев функционирование нарушается из-за контактной неисправности в канальном разъеме (устраняется повторной установкой ТЭЗ в каркас микроЭВМ). Иногда один и тот же ТЭЗ в разных местах каркаса либо в разных каркасах функционирует с различной надежностью. Поэтому перед подключением системы к контуру управления требуется тщательная совместная отладка каждого ТЭЗ в конкретных каркасах микроЭВМ.

Методика значительно упрощает поиск неисправности и восстановление работоспособности системы управления на базе микроЭВМ в условиях эксплуатации.

Телефон для справок о приобретении технической документации и аппаратуры: 113-49-72 (Москва).

ЛИТЕРАТУРА

1. Мини- и микроЭВМ в управлении промышленными объектами / Л. Г. Филиппов, И. Р. Фрейдзон, А. Давыдовичу, Э. Дятку: Пер. с рум. Э. Дятку под общ. ред. И. Р. Фрейдзона, Л. Г. Филиппова. — Л.: Машиностроение, 1984. — 336 с.
2. Каган Б. М., Мкртумян И. Б. Основы эксплуатации ЭВМ / Под ред. Б. М. Кагана. — М.: Энергоатомиздат, 1983.— 376 с.
3. Модули «Электроника МС-3404» (15УЗПП 16КХ16). Техническое описание и инструкция по эксплуатации.

Статья поступила 15 января 1985 г.

РЖ АВТ-86

2Б534. **Системные требования к БИС для графической обработки.**— Compcon spring '85 : 30th IEEE Comput., soc. int. conf. San Francisco, calif., febr. 25—28, 1985. Los alamitos, calif., 1985, 272—275.

Предлагается для разработки недорогостоящих персональных систем (С) графической обработки (ГО) использовать универсальный МП, реализующий большинство функций ГО. Кроме того, этот МП должен выполнять функции графического контроллера, управления буферным ЗУ, синхронизации отображения данных. Для создания С Г0, требующих максимальной производительности, должны использоваться конвейерные мульти-МП С. Они состоят из МП для построения двумерных и 3-мерных проекций, МП для стандартных функций ГО (задание ширины линий, перемещение курсора, заполнение трафарета, затенение изображений и др.), двух МП для построения изображения, МП графического контроллера. МП связаны между собой общей шиной передачи данных.

УДК 681.327.023.:621.391.64

М. Е. Раденко, В. К. Сеньков

РЕАЛИЗАЦИЯ ВОЛОКОННО-ОПТИЧЕСКОГО ИНТЕРФЕЙСА ДЛЯ МИКРОПРОЦЕССОРНЫХ СИСТЕМ

Применение многопроводных системно-ориентированных интерфейсов с электрическими соединителями [1] в распределенных управляющих комплексах на базе микропроцессорных средств сопряжено с определенными трудностями. Емкости монтажа не позволяют при сохранении быстродействия увеличить длину шин интерфейсов. В некоторых случаях возникает необходимость располагать контроллеры ввода-вывода (КВВ) вблизи от внешних устройств или конструктивно совмещать с ними. При этом каналы управления должны обеспечивать заданное быстродействие, низкую вероятность появления ошибок и защиту от воздействия наводимых напряжений.

Применение волоконно-оптических компонентов для передачи информации значительно упрощает решение задачи удлинения интерфейсных шин микропроцессорных систем. Современные волоконно-оптические линии связи (ВОЛС) способны передавать цифровые сигналы без ретрансляции со скоростями до десятков мегабит в секунду на расстояния в несколько километров. Вариант структурной схемы распределенной системы управления с применением элементов ВОЛС представлен на рис. 1.

Один из способов передачи данных по ВОЛС основан на временно́м уплотнении цифровых сигналов с применением специальных самосинхронизирующихся кодов, содержащих информационную последовательность и синхросигналы. Широко распространен для этих целей манчестерский код (код М). Необходимость в применении этого кода [2] возникает из-за бо́льшого числа нулей или единиц подряд в передаваемой информации, что может привести к неоправданному расходу ресурса оптического передатчика.

Порядок преобразования параллельной информации в последовательность кода М поясняется рис. 2,*а*. В коде М (рис. 2,*б*, *в*) каждый двоичный разряд отображается переходом уровней, причем направление перехода определяет значение двоичной переменной. Переходы имеют место в середине отведенного каждому элементу временно́го интервала. Если низкий уровень сменяется высоким, то передается «Лог. 0», а при переходе от высокого уровня к низкому — «Лог. 1». Спектр пе-

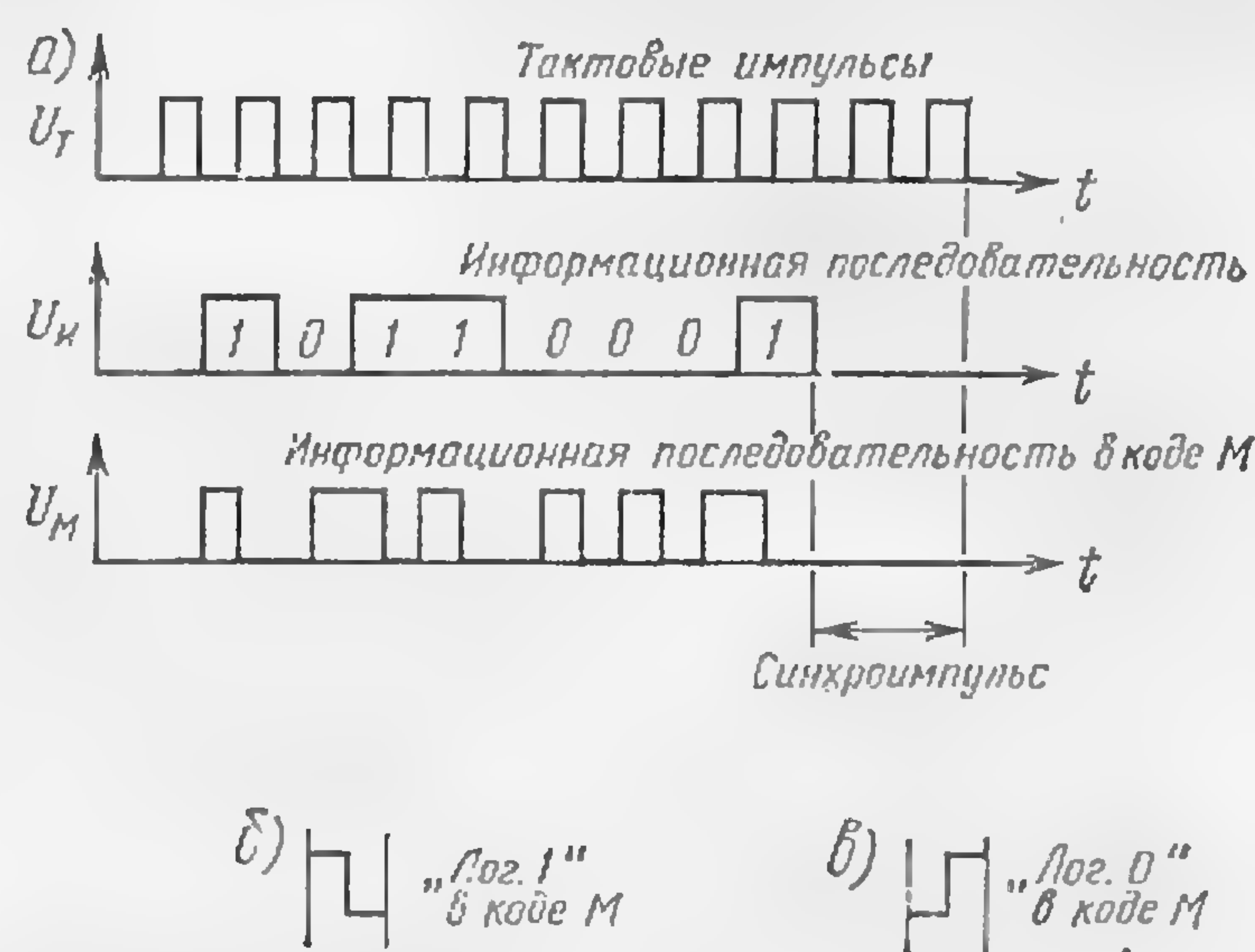


Рис. 2. Преобразование байта $(10110001)_2$ в информационную последовательность в манчестерском коде

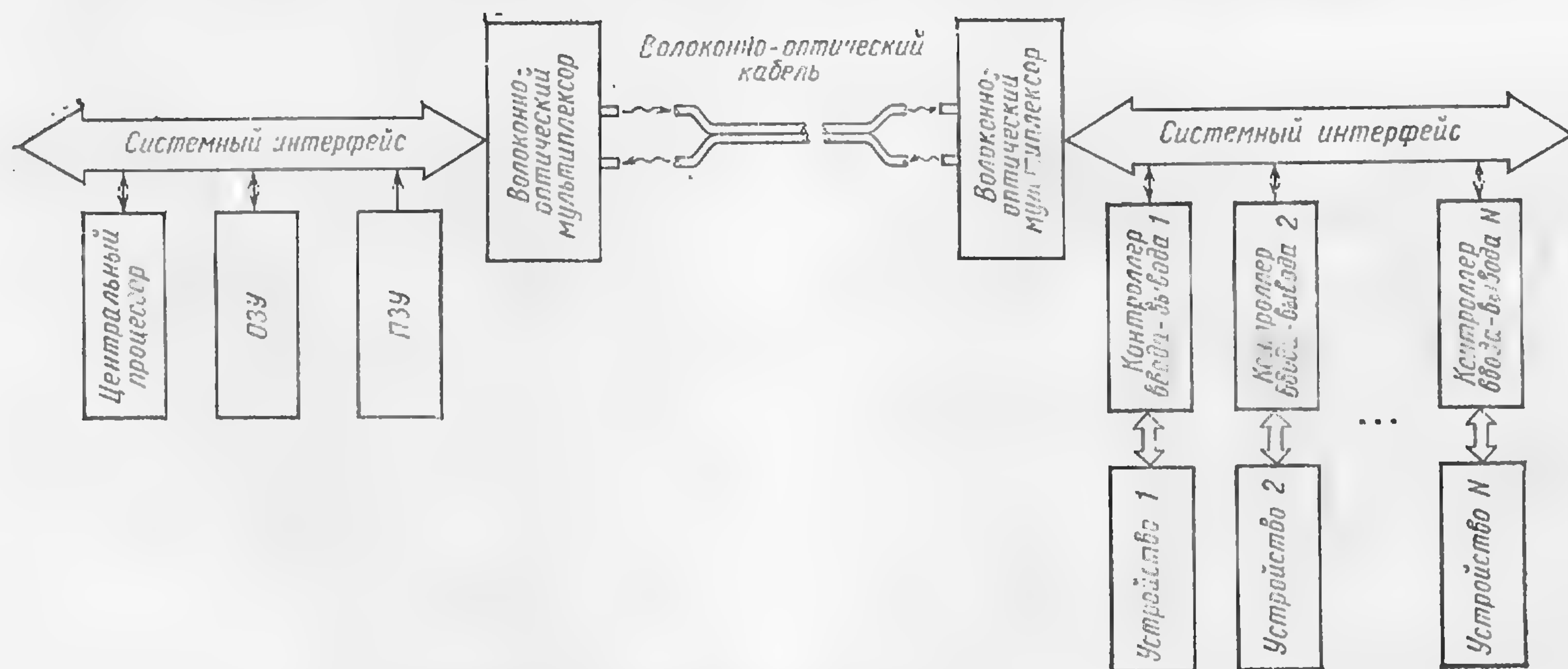


Рис. 1. Структурная схема распределенной системы управления

редаваемой информации может занимать полосу от постоянной составляющей до половины значения тактовой частоты, сигналы в коде М — от половины до полного значения этой частоты. Поскольку отношение граничных частот кода (максимальной частоты спектра к минимальной) представляет собой важный параметр, преимущество этого кода становится очевидным. Все процессы синхронизации для приема информации выполняются в конечном итоге на уровне разрядов, а также на уровне символов (кадров). Для покадровой синхронизации могут использоваться недопустимые разряды манчестерского кода, как показано на рис. 2, поясняющем передачу байта информации с наличием синхроимпульса.

Временно́е уплотнение передаваемых данных и кодовые преобразования выполняют волоконно-оптические мультиплексоры [3] (рис. 3). Предназначенные для передачи электрические сигналы, преобразованные в манчестерский код с помощью параллельно-последовательного регистра и шифратора, поступают в оптический передатчик, который формирует световые импульсы, вводимые в ВОЛС. При работе на прием световые импульсы с ВОЛС преобразуются оптическим приемником в последовательность электрических сигналов в манчестерском коде. Дешифратор выделяет биты информации и тактовые сигналы для записи в сдвиговый регистр. Полученный кадр записывается в выходной буфер в момент прихода синхроимпульса, который вырабатывается дешифратором в конце информационной последовательности.

Системный интерфейс с волоконно-оптическим каналом (рис. 4) содержит три шины: адреса, данных и управления. Для команд ввода и вывода используются восемь младших разрядов шины адреса, 8-разрядная шина данных и два сигнала управления: чтения из устройства ввода (ЧТВВ); записи в устройство вывода (ЗПВВ).

На временно́й диаграмме операции Запись в контроллер вывода (рис. 5) показано взаимодействие сигналов по шинам микропроцессорной системы и устройств вывода. По команде OUT микропроцессор передает на шину адреса соответствующий код по линиям

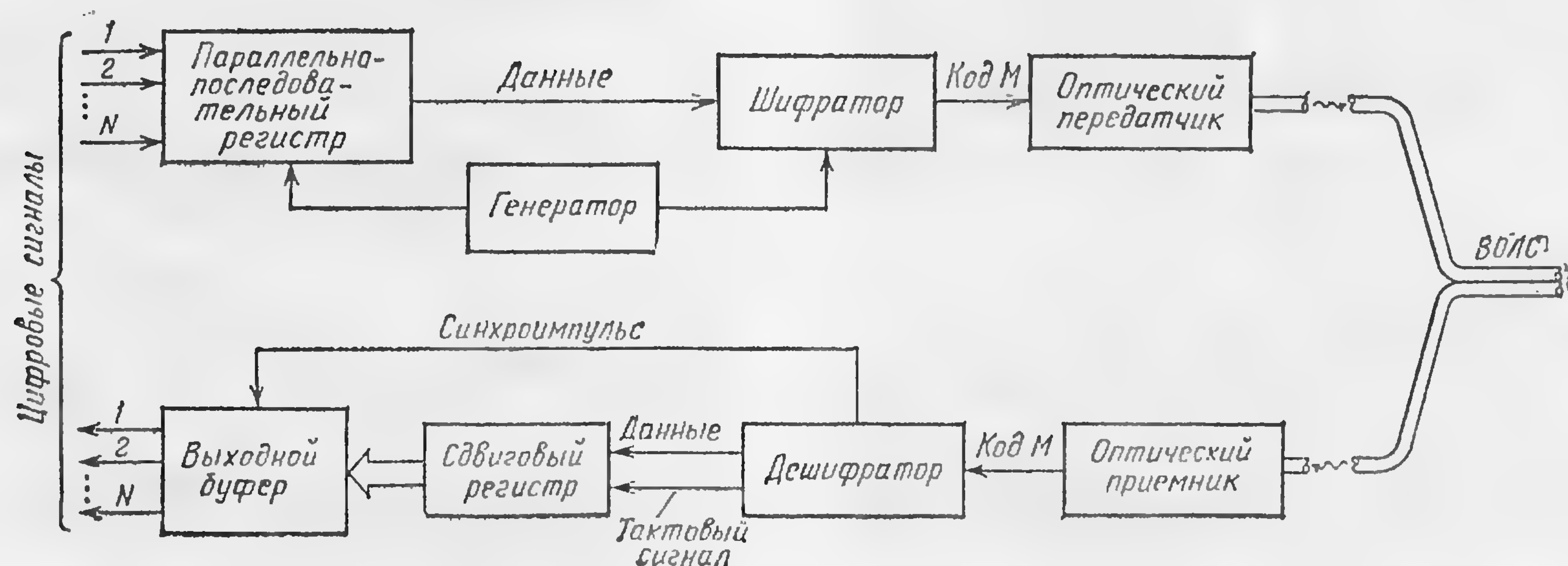


Рис. 3. Структурная схема устройства для передачи и приема цифровой информации (УППЦИ)

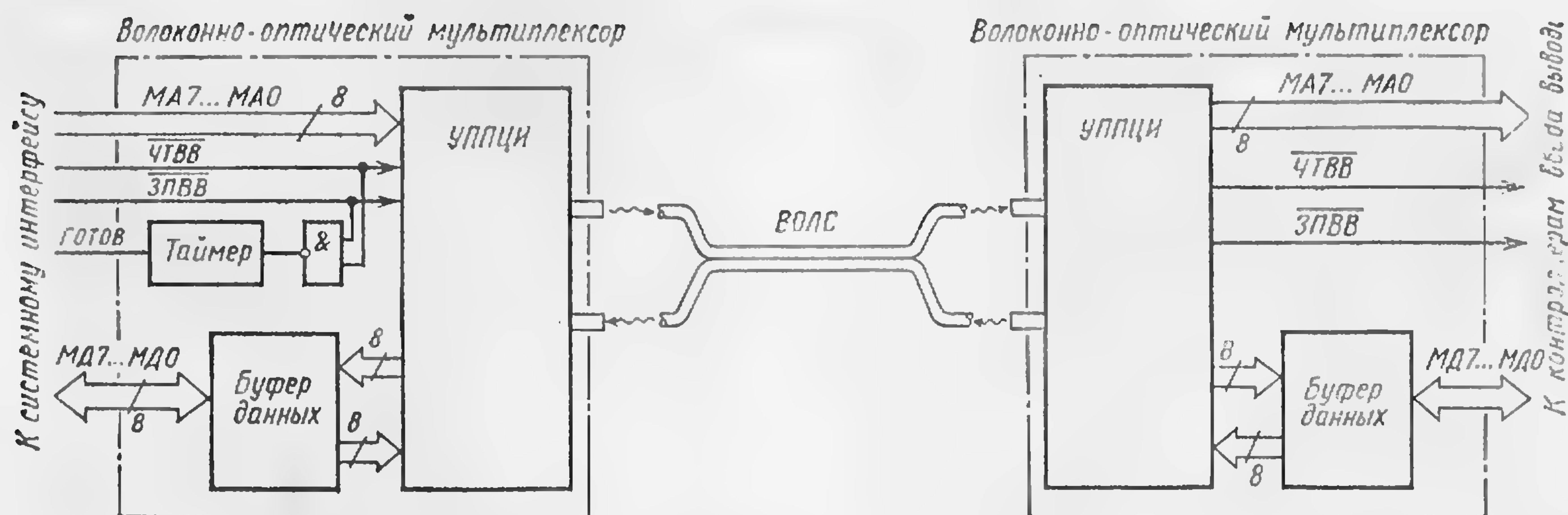


Рис. 4. Структурная схема системного интерфейса с применением волоконно-оптического мультиплексного канала

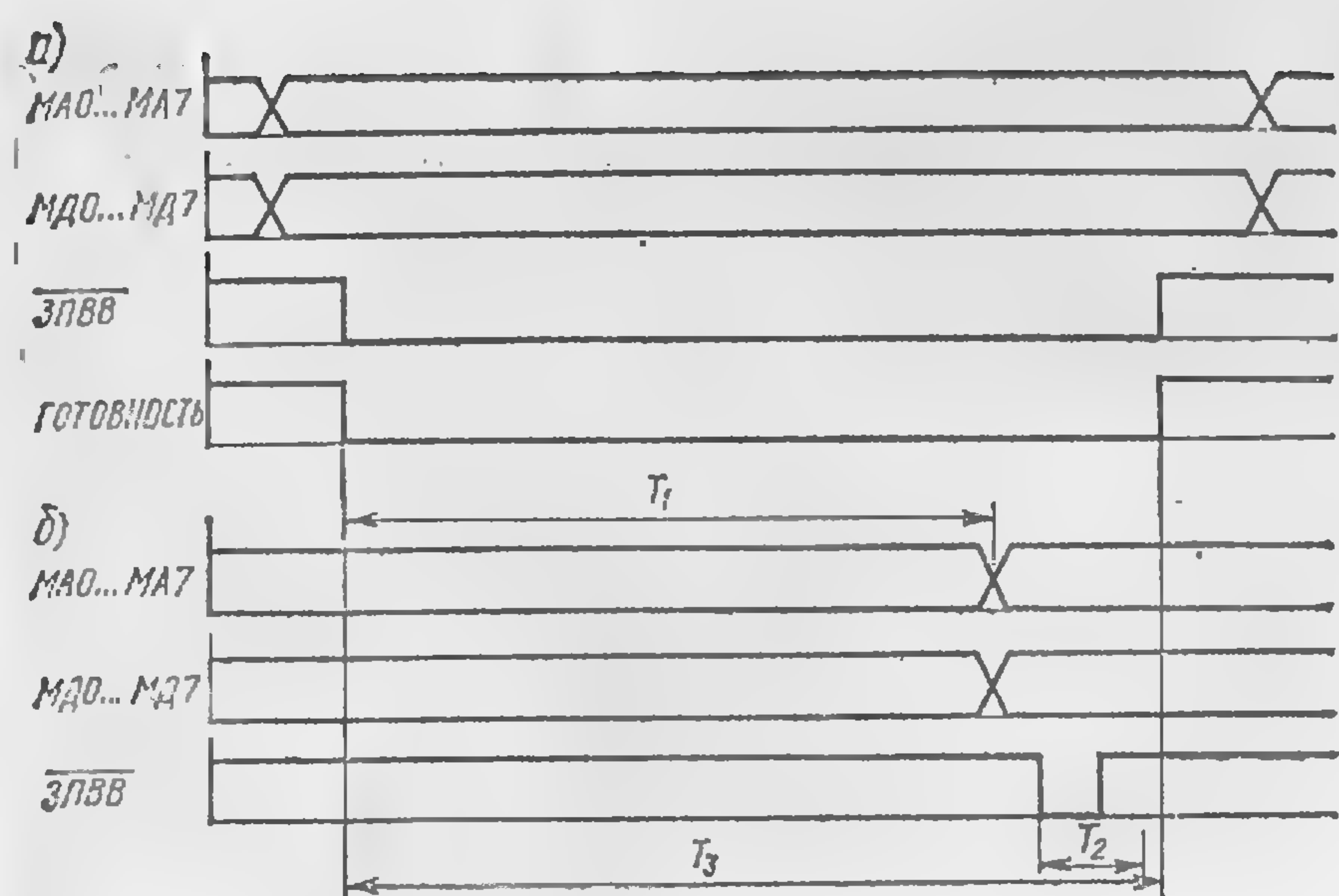


Рис. 5. Временна́я диаграмма операции Запись в КВВ: *а* — на стороне микропроцессорной системы; *б* — на стороне устройств ввода-вывода

МА0...МА7 и информацию по шине данных (МД0...МД7). По сигналу ЗПВВ срабатывает таймер, который переводит БИС КР580ВМ80А в состояние «не готов» на время осуществления операции Запись (T_3). Этим же сигналом запускается устройство для передачи и приема цифровой информации (УППЦИ). На стороне контроллеров аналогичное устройство восстанавливает электрический системный интерфейс. Информация по шинам адреса, данных и сигнал ЗПВВ фиксируются соответствующим контроллером вывода. Промежуток T_1 определяет время передачи по ВОЛС кадра информации, а T_2 — время осуществления записи в КВВ. По окончании операции Запись таймер восстанавливает готовность микропроцессора.

Временны́е диаграммы операции Чтение из устройства ввода приведены на рис. 6. По команде IN микропроцессор и волоконно-оптические мультиплексоры формируют на стороне контроллеров ввода-вывода информацию по шинам адреса, данных и сигнал ЧТВВ. Соответствующий контроллер ввода за период времени T_2 передает байт данных по обратному каналу волоконно-оптических мультиплексоров. Через промежуток времени T_3 данные поступают на линию МД0...МД7 системного интерфейса со стороны микропроцессора. При выполнении этой операции таймер также снимает и восстанавливает готовность микропроцессора по истечении времени T_4. Таймер настраивается на максимальное время осуществления операции Чтение из КВВ.

При скорости передачи данных по ВОЛС 8 Мбит/с максимальная задержка передачи информации по интерфейсу к удаленным контроллерам составляет 2,6...4,0 мкс. Дополнительная задержка распространения оптического сигнала в ВОЛС около 5 мкс/км. Реализация волоконно-оптического интерфейса не сопряжена с большими аппаратными и технологическими затратами. Современная отечественная компонентная база позволяет разместить волоконно-оптический мультиплексор на плате размерами не более 150×150 мм.

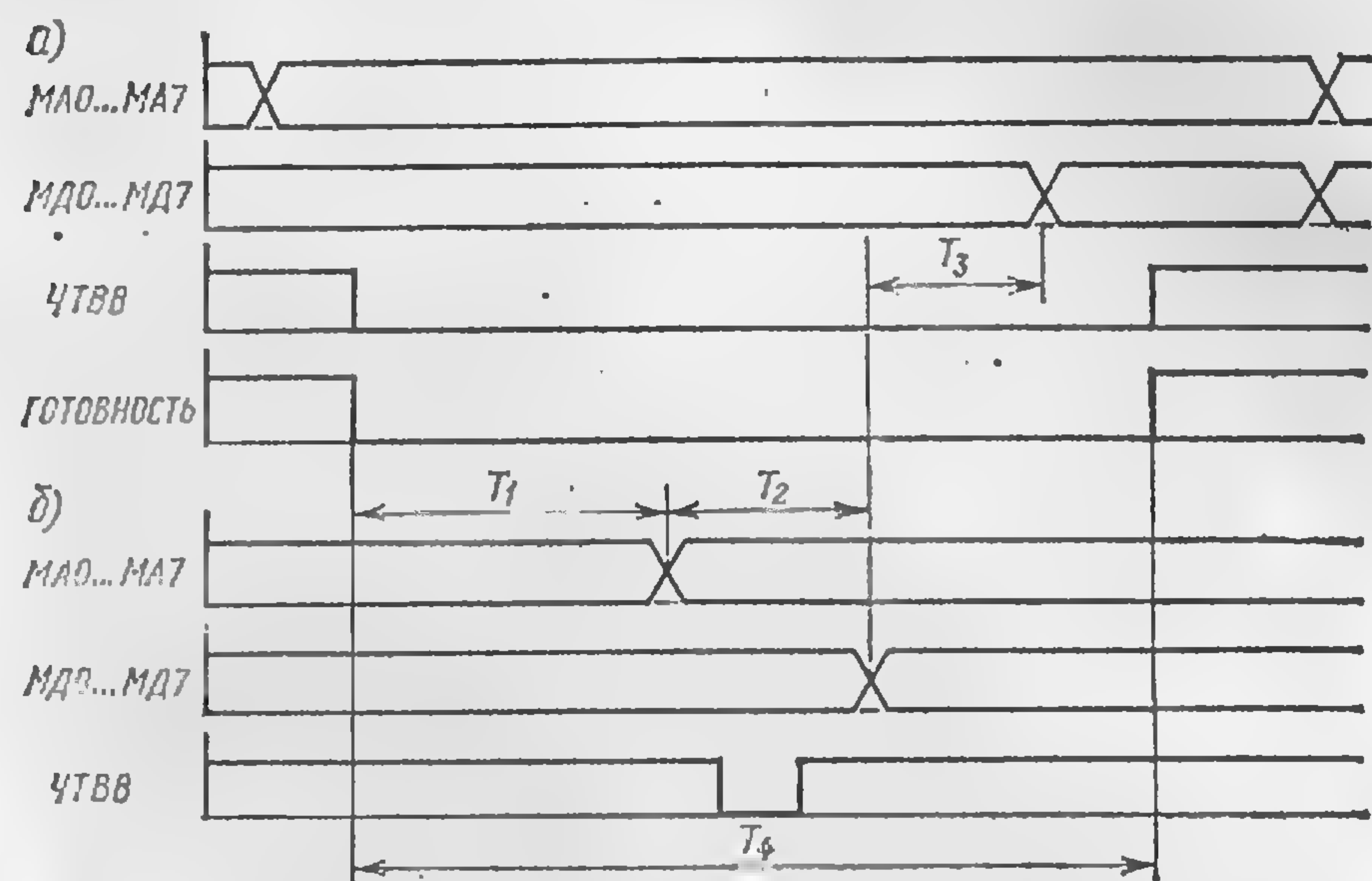


Рис. 6. Временна́я диаграмма операции Чтение из КВВ:
а — на стороне микропроцессорной системы; *б* — на стороне устройств ввода-вывода

В заключение следует отметить, что в качестве УППЦИ волоконно-оптического мультиплексора (см. рис. 4) могут быть применены серийно выпускаемые изделия «Электроника МС 4101» [4].

ЛИТЕРАТУРА

1. Мячев А. А., Никольский О. А. Стандартные интерфейсы микропроцессорных систем. — Микропроцессорные средства и системы, 1984, № 1, с. 27—33.
2. Сандерс Л. Однокристальный кодекс для преобразования кода без возвращения к нулю в манчестерский код. — Электроника, 1982, № 15.
3. Ломбарди В. Р. Волоконно-оптический мультиплексор на 16 каналов. — Электроника, 1982, № 6.
4. Ананян М. А., Мельникова О. В. Оптоволоконные локальные сети. — Микропроцессорные средства и системы, 1984, № 4.

Статья поступила 20 апреля 1985 г.

УДК 681.7.068:681.327.21/.22

В. Д. Баронец, Р. Н. Бойцов, А. Г. Воробьев, Г. И. Забегалова

УСТРОЙСТВО СВЯЗИ МИКРОЭВМ С ОБЪЕКТАМИ УПРАВЛЕНИЯ ПО ВОЛОКОННО-ОПТИЧЕСКИМ ЛИНИЯМ

Применение оптоволоконных линий передачи информации [3] в системах на основе микроЭВМ требует создания дополнительных устройств сопряжения. Для обеспечения информационной и конструктивной совместимости с микроЭВМ, оперативной реконфигурации микропроцессорных систем в эксплуатационных условиях разработаны устройства связи с объектами по волоконно-оптическим кана-

лам — модули приемопередающий и интерфейсные.

Модуль приемопередающий «Электроника МС 4605» обеспечивает подключение ВУ, имеющего последовательный интерфейс ИРПС, к волоконно-оптическому кабелю (ВОК) (рис. 1). Техническая характеристика модуля приведена ниже.

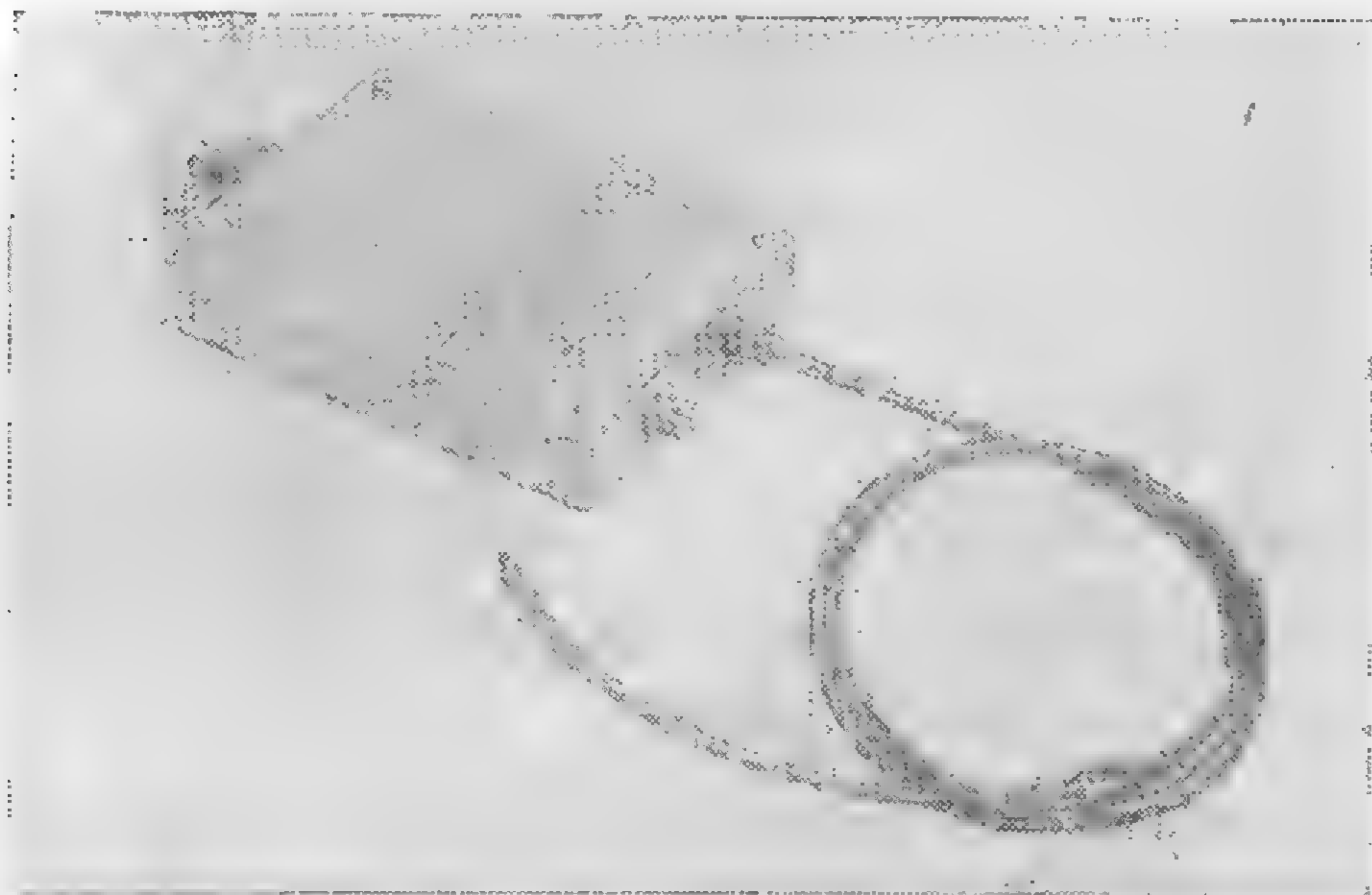


Рис. 1. Модуль приемопередающий

Скорость обмена, Кбит/с, не более	100
Длина волны излучения, мкм	0,95
Чувствительность приемника, Вт	10^{-7}
Напряжение внешнего источника питания, В	$+12\pm\frac{5}{2}$ %
Потребляемая мощность, Вт	2
Диапазон рабочих температур, °С	—10... +50
Соединение с ВОК	разъемное
Длина ВОК, м, не более	500
Конструкция	встраиваемая
Масса, кг	0,25
Габаритные размеры, мм	135×85×20
Цена, руб.	200

В основе работы модуля лежит принцип преобразования оптических сигналов инфракрасного диапазона излучения, распространяемых по ВОК, в электрические и электрических сигналов в оптические. Модуль состоит из двух узлов: приемника и передатчика (рис. 2).

Приемник оптических сигналов включает фотодиод VD2, преобразователь ток — напряжение DA1, усилитель напряжения DA2, компаратор DA3 и согласователь DD1.1. Инвертирующий усилитель DA1 обладает малым входным сопротивлением. Коэффициент усиления неинвертирующего усилителя DA2 регулируется изменением величины резистора R6*, включенного в цепь обратной связи усилителя. Каждое значение сопротивления резистора R6* соответствует определенному диапазону длины ВОК (при этом для данного диапазона обеспечивается независимость скважности принимаемых сигналов от их мощности). Необходимость изменения коэффициента усиления связана с использованием ВОК различной длины и потерь. Порог срабатывания DA3 с помощью регистров R9, R10 выставлен таким образом, чтобы скомпенсировать собственные шумы приемника; согласователь DD1.1. обеспечивает соответствие нагрузок модуля и внешнего устройства.

Передатчик состоит из формирователя импульсов DD1.2 и излучателя VD4.

В качестве излучателя выбран светодиод АЛ107Б, который позволяет при токе накачки 100 мА излучать оптическую мощность 10 мВт. При использовании ВОК типа кварц-кварц диаметром 125/50 мкм и светодиода АЛ107Б потери при их стыковке 30 дБ [4]. С учетом этих потерь введенная в ВОК оптическая мощность составляет 10 мкВт, и ее достаточно, чтобы при чувствительности приемника 10^{-7} Вт и соотношении сигнал/шум 6/1 обеспечить передачу сигналов на расстояние до 500 м с затуханием до 20 дБ/км. Кроме того, светодиод АЛ107Б является недорогим комплектующим изделием и благодаря своей конструкции удобен для установки в оптический соединитель.

Для развязки приемника и передатчика по цепи питания применяется П-образный фильтр L, C12, C13. При отсутствии сигнала в линии связи напряжение в ней соответствует

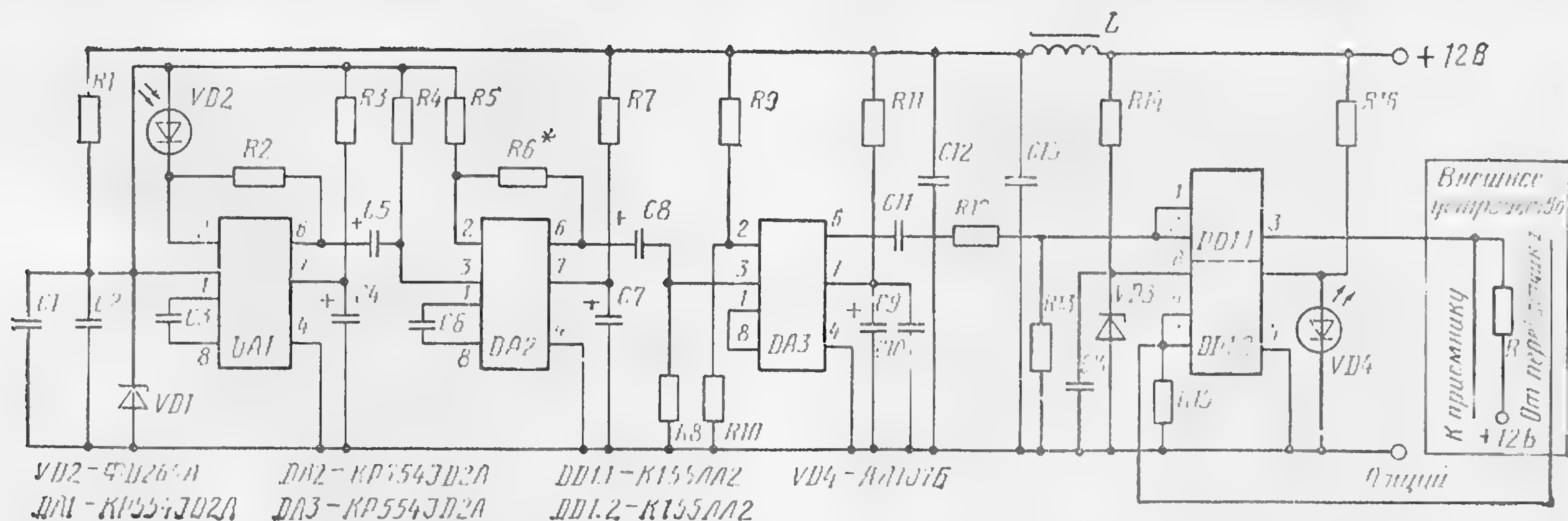


Рис. 2. Принципиальная схема модуля приемопередающего

состоянию «Лог. 0» (0,5 В), при наличии сигнала — «Лог. 1» (2,5 В). Обмен с ВУ по последовательному каналу организован в соответствии с требованиями НМ МПК по ВТ10—78 СМ ЭВМ «Интерфейс для радиального подключения устройств с последовательной передачей информации» и ОСТ 11 305.916. Модуль, представляющий собой двустороннюю печатную плату, заключен в стальной корпус, выполняющий роль электромагнитного экрана. На передней панели корпуса устанавливаются оптические соединители для подключения ВОК, на задней панели — разъем для связи с ВУ.

Модули интерфейсные «Электроника МС 8801», «Электроника МС 8802» осуществляют по ВОК двустороннюю связь между двумя ВУ, с последовательными интерфейсами ИРПС и ЭВМ с параллельным интерфейсом МПИ (рис. 3).

Работа модуля основана на преобразовании параллельного кода, принятого из микроЭВМ, в последовательный в виде оптических сигналов и передаче их по ВОК к удаленному ВУ, а также приеме последовательного кода в виде оптических сигналов, переданных по другому ВОК ВУ, обратном преобразовании их в параллельный код и передаче последнего в микроЭВМ.

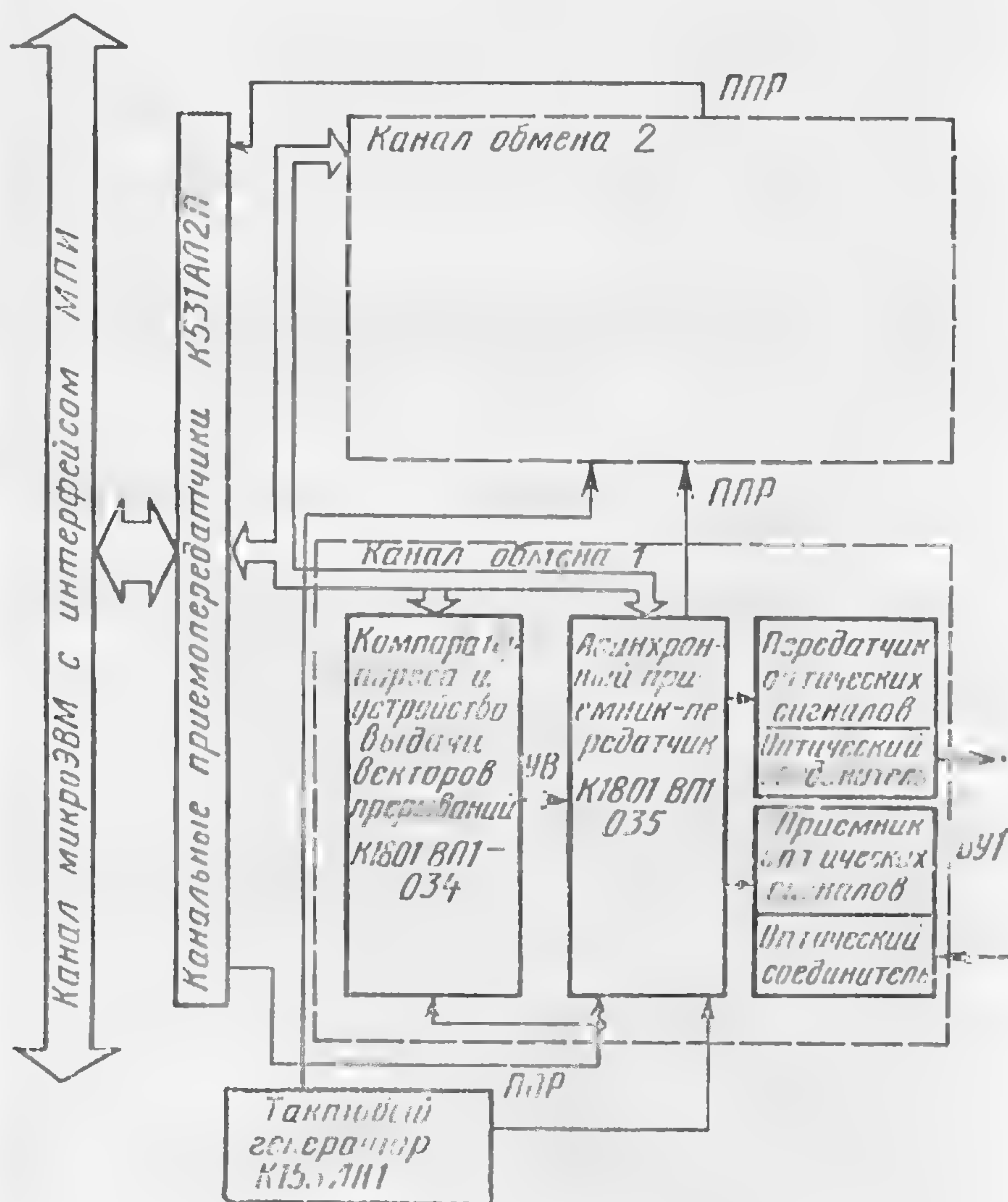


Рис. 3. Структурная схема модуля интерфейсного

Техническая характеристика модулей приведена ниже.

Число каналов обмена	2
Режим обмена	программный, по прерыванию
Интерфейс связи с ЭВМ	МПИ
Скорость обмена, бит/с	75, 150, 300, 600, 1200, 2400, 4800, 9600, 19 200, 57 600
Установка адресов регистров и векторов прерывания	произвольная и независимая по каждому каналу
Длина волны излучения, мкм	0,95
Чувствительность приемника, Вт	10^{-7}
Потребляемый ток, А	
+5 В ±5 %	1,1
+12 В ±5 %	0,025
−12 В ±5 %	0,015
Диапазон рабочих температур, °С	−10...+50
Соединение с ВОК	разъемное
Длина ВОК, м, не более	500
Масса, кг	0,2
Габаритные размеры, мм	252×143×12— МС 8801; 144,45×220××12— МС 8802
Цена, руб.	600

Для связи модуля с ВУ используется старт-стопный метод передачи, в котором знаки передаются как отдельные элементы и могут отделяться произвольными временными интервалами. В начале каждого знака вводится стартовая посылка длительностью 1 бит. Чтобы обеспечить переход от предыдущего знака к последующему, каждому знаку придается стоповая посылка 1 или 2 бита.

Модуль состоит из стабилизированного кварцем тактового генератора на 4,608 МГц, канальных приемопередатчиков для связи с каналом микроЭВМ и двух каналов обмена с ВУ через ВОК. Каждый канал обмена состоит из компаратора адреса и устройства выдачи векторов прерываний, специального асинхронного приемопередатчика, являющегося однокристальным контроллером ВУ, работающих на линию связи с последовательной передачей информации в дуплексном режиме, приемника и передатчика оптических сигналов, оптических соединителей. Устройство и работа каждого из двух независимых каналов обмена модуля с ВУ идентичны, поэтому рассмотрим работу одного канала.

БИС К1801ВП1-035, на основе которой построен асинхронный приемопередатчик, преобразует параллельную информацию в последовательную и наоборот в соответствии с требованиями ОСТ 11 305.903—80 «Межмодульный параллельный интерфейс». БИС содержит четыре программно-адресуемых регистра и два источника адресов векторов прерыва-

ния. Адреса регистров и адреса векторов сменные по группам. Группы адресов регистров и векторов прерываний выбираются заданием комбинации уровней напряжения на выводах БИС. Для внешнего произвольного набора адресов регистров и внешнего источника адреса вектора прерывания БИС выбрана четвертая группа.

Внешний произвольный набор адресов регистров и векторов прерываний осуществляется с помощью БИС К1801ВП1-034, работающей в режиме компаратора адреса и устройства выдачи векторов прерываний. Адрес, необходимый для сравнения, устанавливается на выводах БИС с помощью перемычек. В случае совпадения заданного адреса с поступающим из микроЭВМ БИС вырабатывает сигнал «Устройство выбрано», который поступает на БИС К1801ВП1-035. Регистры БИС выбираются по адресу, состоящему из первого и второго разрядов. Таким образом происходит обращение микроЭВМ к любому из четырех программно-адресуемых регистров БИС.

Адрес вектора прерывания устанавливается с помощью перемычек на выводах БИС К1801ВП1-034. Во время работы модуля в режиме прерывания программы ВУ, подключенное к первому каналу оптического соединителя модуля, имеет наивысший приоритет в группе устройств, подключенных к одному модулю, за ним следует второй канал. Это обеспечивается распространением сигнала «Предоставление прерывания» (ППР) последовательно от канала обмена 1 к каналу 2.

Режим работы канала обмена устанавливается перемычками на выводах БИС К1801ВП1-035:

скорость обмена;

формат посылки (форматы посылок 7,8 бит);

число СТОП-бит в посылке;

работа с паритетом или без паритета, а также работа с паритетом по четности или по нечетности (формирование бита четности и контроль четности, формирование бита нечетности и контроль нечетности, нет формирования бита паритета и нет контроля паритета).

Выводы БИС, предназначенные для соединения с ВУ (аналогичны описанным в модуле приемопередающем, приведенном на рис. 2), подключаются к DD1.1 (вывод 3) и DD1.2 (выводы 6, 7) соответственно приемника и передатчика оптических сигналов. Приемник и передатчик заканчиваются оптическими соединителями, к которым подключается ВОК для связи с ВУ.

Модули МС 8801 и МС 8802 отличаются друг от друга габаритными размерами и видами разъемных соединений.

При использовании модуль приемопередающий встраивается в объект управления ВУ, а модули интерфейсные устанавливаются в крейты соответствующих микроЭВМ.

Предусматривается возможность проверки пользователем правильности функционирования модуля интерфейсного. Для этого он устанавливается в крейт с микроЭВМ, и выходы передатчиков с входами соответствующих приемников в каналах модуля соединяются проверочным ВОК. Каждый канал модуля имеет четыре программно-адресуемых регистра: ввода, состояния ввода, вывода, состояния вывода и два адреса вектора прерывания по вводу, по выводу. Это позволяет проверить модуль по любой программе как в программном режиме работы, так и в режиме прерывания программы. Пользователю предлагается простая тестовая программа, по которой произвольный символ выводится из микроЭВМ через регистр вывода выбранного канала модуля, затем вводится в микроЭВМ через регистр ввода этого же канала и далее выводится на экран дисплея.

При необходимости замены ВОК (в случае его неисправности, изменения требований к длине кабеля и др.) пользователь может самостоятельно собрать оптический соединитель. Он состоит из розетки, куда устанавливается светодиод АЛ107Б или фотодиод ФД265А, и вилки, представляющей собой металлическую трубку с внутренним диаметром 160 мкм, в которую вклеивается световод ВОК. Поверхность торца световода формируется путем скола скрайбером выступающей из трубки части световода, при этом юстировку световода в оптическом соединителе нет необходимости производить из-за большой разницы в диаметрах излучателя (приемника) оптических сигналов и световода. Процесс сборки оптического соединителя контролируется с помощью микроскопа МБС-2.

Модули прошли этапы испытания, внедряются, имеется конструкторская документация для их изготовления.

За справками обращаться по тел. 65-85-36 (г. Горький).

ЛИТЕРАТУРА

1. Баронец В. Д., Бачкова Т. У., Соломин В. Н. Устройства связи микроЭВМ «Электроника 60М» с объектами управления. — В кн. Электронная техника. Сер. 7, ТОПО, вып. 1, 1983, с. 19—21.
2. Баронец В. Д., Клюшин Б. А., Пряхин Л. П. Иерархический микроУВК для АСУ производственными процессами. — Электронная промышленность, 1983, вып. 3, с. 60—62.
3. Ананян М. А., Мельникова О. В. Оптоволоконные локальные сети. — Микропроцессорные средства и системы, 1984, № 4, с. 32—36.
4. Storozum S. L. Estimating the power coupled into an optical fiber.— Electronics, 1980, № 4, p. 32—36.

Статья поступила 1 августа 1985 г.

УДК 519.683.5

А. К. Дудоладов

СИСТЕМА АВТОМАТИЗАЦИИ ПОДГОТОВКИ ШАХМАТИСТОВ «ДЕБЮТ»

Одно из традиционных направлений в области искусственного интеллекта — разработка шахматных алгоритмов и программ. Над этой проблемой работают многие коллективы математиков в разных странах, создано немало сильных игровых программ. Шахматы являются удобной моделью при решении различных сложных задач [1].

Однако для шахматистов высокой квалификации эти разработки представляют ограниченный интерес: общий уровень игровых программ остается значительно ниже уровня игры гроссмейстера, и использование шахматных программ при подготовке шахматистов сводится в основном к решению некоторых задач и анализу отдельных эндшпилей.

Основная цель разрабатываемой системы состоит в создании средств, облегчающих теоретическую подготовку к соревнованиям шахматистов различной квалификации. Значение теоретической подготовки в современных шахматах постоянно возрастает, и квалифицированные шахматисты сталкиваются с рядом трудностей при обработке большого потока информации [2]. Использование информационно-поисковой системы (ИПС) значительно облегчает эту работу.

Технические средства современных персональных компьютеров, такие как цветная графика, твердый диск большой емкости, устройство ввода типа «мышь» и др., позволяют создавать программы, значительно облегчающие работу, непосредственно связанную с анализом партий и позиций.

Автоматическая классификация партий

Все партии перед выходом в печать или при занесении в ИПС классифицируются по специальным шахматным индексаторам. Цель этой классификации состоит в более точной идентификации дебютного варианта. Именно индекс является во многих случаях главным признаком при поиске партий. Автоматическая индексация партий, реализованная в системе, основана на классификации Рабара.

Разработка алгоритма индексации связана с рядом трудностей, возникающих при формализации принципов классификации партий [3]. В основе алгоритма индексации лежит возможность представления классификатора древовидной структурой: ребрам дерева приписаны полуходы, вершинам соответствуют индексы (рис. 1).

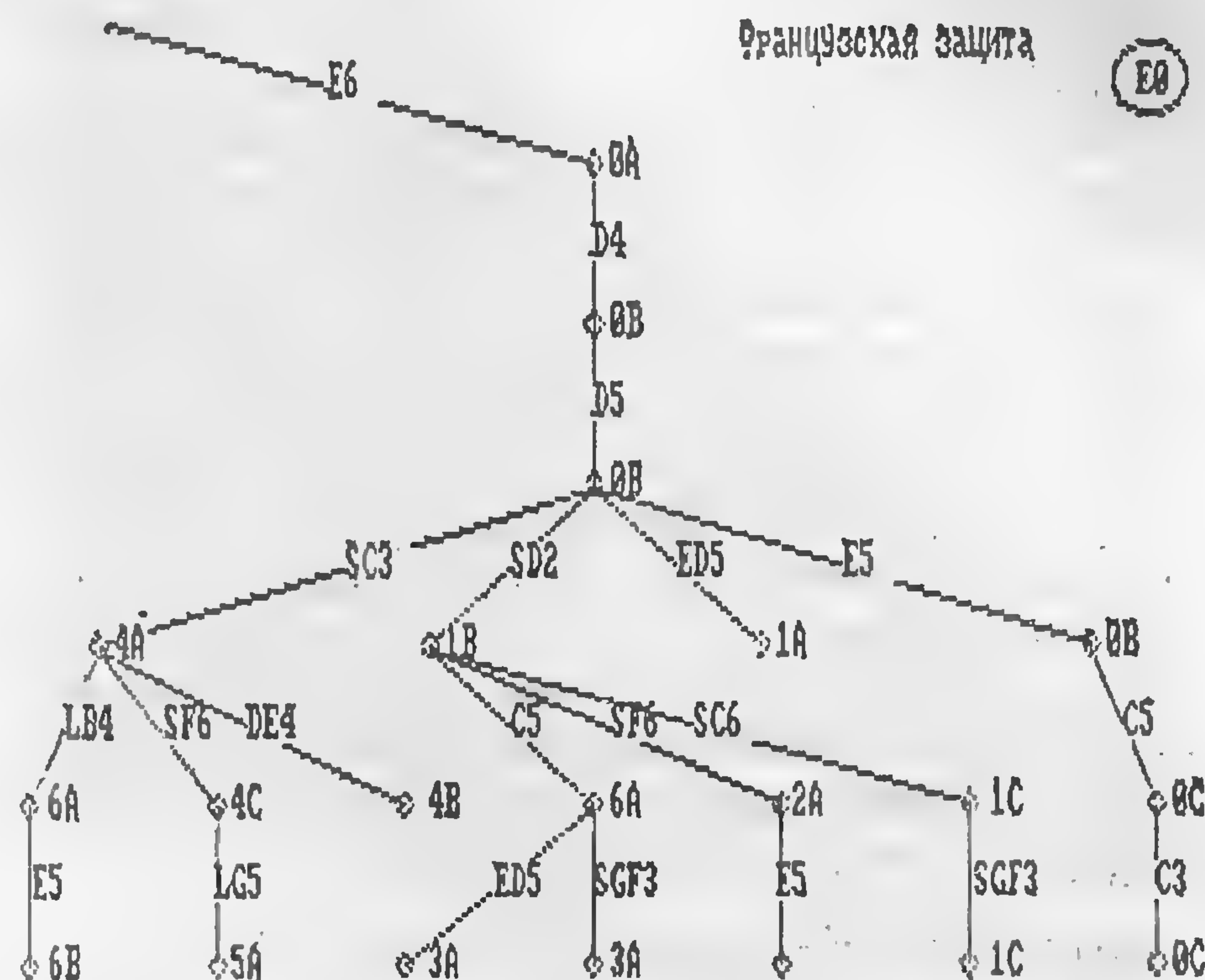


Рис. 1. Часть дерева вариантов для раздела «Французская защита»

Первый этап алгоритма индексации — поиск индексов-кандидатов — состоит в построении некоторого подмножества индексов, наиболее соответствующих варианту, разыгранному в партии. Определим алгоритм с помощью индукции по ярусам дерева:

1. Базис индукции. Просматриваются все полуходы первого яруса, и осуществляется поиск каждого из них в части партии, ограниченной допустимой глубиной перестановки ходов. Запоминаются те номера вершин первого яруса, в которые ведут ребра с найденными в партии ходами.

2. Индуктивный переход. От предыдущего шага имеем массив. В нем находятся указатели тех вершин, пути в которые уже найдены. Далее, для каждого из ребер, выходящих из этих вершин, производится сравнение аналогично базису индукции. Алгоритм заканчивает свою работу, если в некотором ярусе нет ни одного совпадения ходов.

Второй этап алгоритма состоит в определении единственного индекса и названия дебюта. Для ее решения используется ряд эвристических критериев.

Поиск индексов-кандидатов демонстрируется построением дерева рассматриваемых вариантов. Пользователь может управлять алгоритмом индексации, изменяя параметры в эвристических критериях и допустимую глубину перестановки ходов. Таким образом, для одной и той же партии можно получать различные множества индексов-кандидатов (де-

рево рассматриваемых вариантов меняет свою глубину и ширину) и различные конечные индексы партии. Это удобное средство, позволяющее шахматисту-индексатору исследовать различные варианты алгоритма.

Дерево классификатора представлено иерархическим списком с записями переменной длины. Каждый элемент списка содержит полуходы и соответствующие им индексы, а также ссылки на следующие вершины дерева. Количество ссылок определяется ветвлением в данном узле. Реализация списка в файле прямого доступа с постоянной длиной записи оказалась несколько неэффективной в смысле использования внешней памяти. Однако, учитывая достаточно быстрый поиск, который обеспечивает эта структура, можно, видимо, признать данный компромисс приемлемым.

При построении дерева рассматриваемых вариантов координаты вершин каждого яруса рассчитываются с помощью равномерного деления по числу вершин (рис. 1).

Информационно-поисковая подсистема

В подсистеме различаются два вида объектов: партии и позиции. Каждая партия в базе данных содержит информацию о фамилиях соперников, результате, наименовании соревнования, годе его проведения. При вводе новой фамилии шахматиста система запрашивает и страну, которую он представляет. Кроме того, сохраняется также индекс, получаемый при автоматической индексации партии.

Рассмотрим ввод информации в систему. Для облегчения и ускорения ввода текста партий разработаны следующие средства:

каждый полуход проходит синтаксический (правильность записи хода) и семантический (соответствие хода шахматным правилам) контроль. Неправильные ходы не вводятся в систему;

текст партии вводится прямо с листа без дополнительных клавиш. Нумерация ходов автоматическая;

ввод очередного хода демонстрируется изменением позиции на экране.

Тем не менее ввод информации остается трудоемкой операцией. При эксплуатации различными пользователями ИПС на многих машинах затраты на ввод заметно сокращаются, так как имеется возможность обмена информацией.

При выводе партий пользователь формирует запрос, в котором может указать произвольную, имеющую смысл комбинацию ключей. Система поддерживает около 100 различных видов запросов. Самый простой и наиболее быстрый способ вызова партии или позиции — загрузка по номеру в каталоге. После того, как партия найдена в системе, ее можно вывести на печать или рассмотреть с помощью разработанных средств анализа.

Важной особенностью является возможность сохранять произвольную позицию на доске в любой момент анализа или игры. При этом указываются либо фамилии шахматистов, в партии которых данная позиция возникла, либо некоторое произвольное имя, что удобно для сохранения позиций этюдов.

ИПС построена на основе инвертированных файлов, что обеспечивает достаточно быстрый поиск. Разработаны средства просмотра и редактирования информации по каждому из каталогов системы. Специальная кодировка текста партии позволяет размещать полуход в двух байтах памяти.

Средства анализа партий и позиций

При анализе партий и позиций активно используются функции ИПС и представления шахматных диаграмм, реализованные графическими средствами (см. вкладку). Неудобства, возникающие при обычном анализе партий и позиций, связаны прежде всего с необходимостью работы с двумя объектами: позицией на доске и текстом партии в журнале. Например, при анализе партии шахматист теряет некоторое время на преобразование текстового представления ходов в их действительное представление на доске. При анализе позиции рассматриваемые на доске варианты необходимо записывать в текстовом виде.

При автоматическом разыгрывании партии шахматист не отвлекается на «чтение» партии и перестановку фигур. Во время разыгрывания можно ускорить, замедлить, приостановить игру, вернуть назад несколько последних ходов. Если партия была записана с комментариями, то при разыгрывании комментарии автоматически выводятся на вторую доску (рис. 2) и проигрываются аналогично.

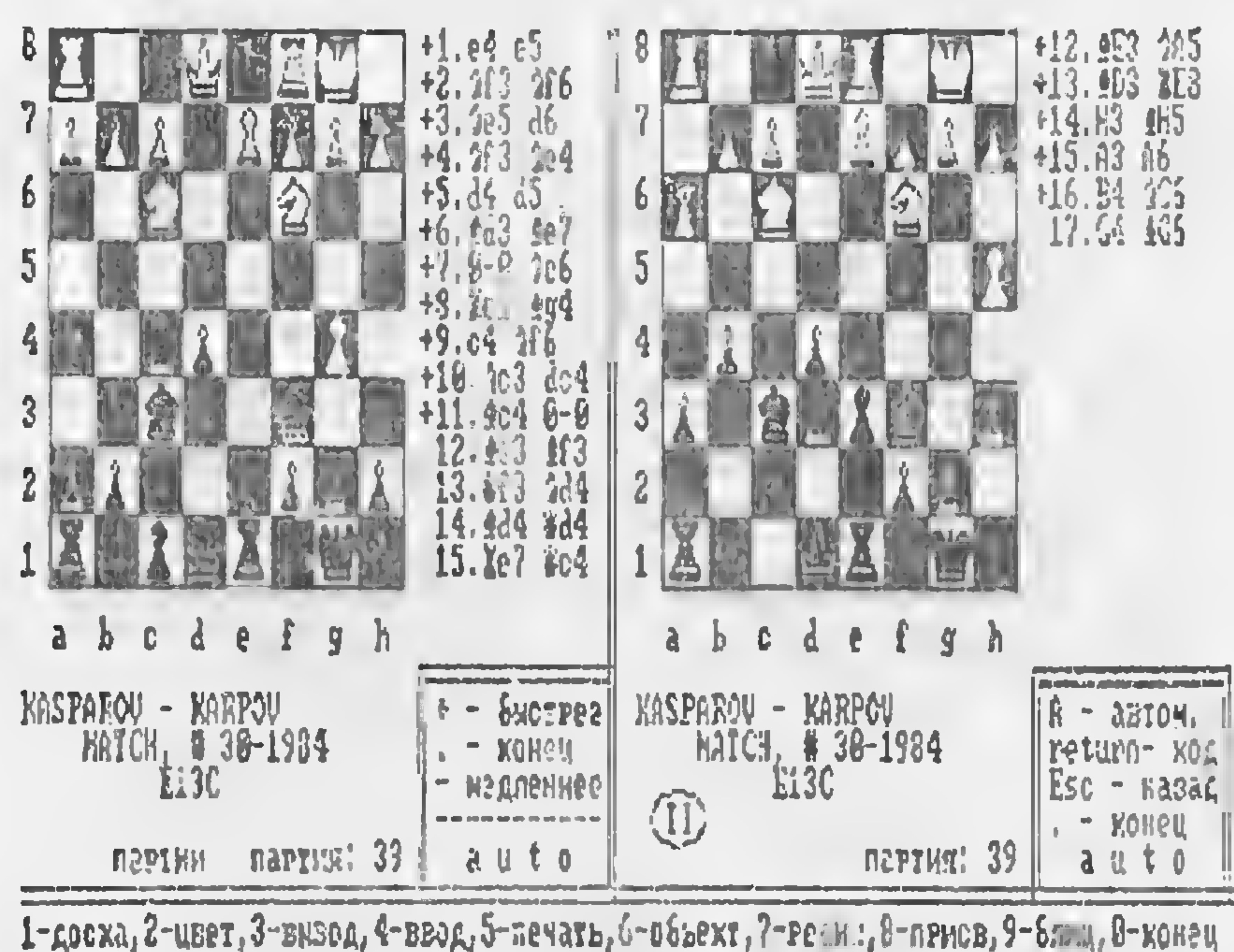


Рис. 2. Анализ позиции с использованием комментариев в режиме двух диаграмм

Устройство типа «мышь» позволяет имитировать взятие и постановку фигуры. При вводе хода нужно, двигая устройство по столу, подвести курсор к выбранной фигуре, «взять» ее с помощью нажатия правой кнопки устройства, переместить курсор на нужное поле и «поставить» фигуру с помощью левой кнопки. Клавиатура и устройство «мышь» работают в режиме параллельного ввода. При вводе ходов манипулятором «мышь» генерируется их текстовое представление и далее можно работать с текстом партии.

При анализе позиции каждый рассмотренный вариант записывается в текстовом виде, и затем можно автоматически повторить все рассмотренные ранее варианты.

Система поддерживает одновременную работу с двумя партиями или позициями (см. вкладку). Это средство предназначено для анализа комментариев, сравнительного анализа партий или позиций. Например, на одной доске можно расставить некоторую позицию, а на другую вывести партию из каталога. В течение нескольких секунд машина может проиграть партию и, если указанная позиция встретится при разыгрывании, укажет номер хода, после которого это произошло.

Пользуясь манипуляторами «мышь», два соперника могут сыграть, например, партию в блиц. Преимущества такой игры следующие: ведется запись текста партии, что позволяет сделать по окончании игры ее анализ; автоматическое переключение часов соперников; нельзя сделать неправильный ход, так как некорректные ходы не вводятся в машину; партию легко отложить, сохранив текущую позицию; не возникает недоразумений по поводу положения той или иной фигуры, падения фигур и т. д.

Рассмотрим реализацию диаграмм шахматных позиций. Используются два набора по шесть фигур для большой и малой диаграмм. Фигуры рисуются специальной программой FIGURE с помощью операторов языка DRAW и запоминаются в файлах. При вызове некоторой программы, использующей позиции, фигуры загружаются из соответствующего файла, который должен находиться в текущем каталоге. В матрице POS (8,8), содержащей коды фигур на доске, позиция имеет внутреннее представление, которое обеспечивает простое отображение ходов на графической диаграмме.

Диалог в подсистеме организован с помощью функциональных клавиш и позиционных меню. Использование 10 функциональных клавиш для основных функций системы позволяет осуществлять быстрые переходы из одного процесса в другой, избегая промежуточных состояний.

Обучение дебютам

Обучение дебютам, реализованное в системе, построено на основе дебютных вариантов классификатора Рабара. На первом этапе инициатива полностью передается пользователю. Он указывает интересующий его дебют из приводимого списка. Далее на экран выводится и последовательно разыгрывается фиксированная часть дебюта. Затем пользователь может рассматривать различные варианты, выбирая тот или иной ход в приводимом меню (см. рис. 2). Если дебютный вариант исчерпан, то указывается его индекс. Рассмотренный вариант можно повторить. Таким образом, одна из функций подсистемы обучения состоит в том, чтобы предоставить пользователю возможность получения справочной информации о том или ином варианте и его индексе.

Путем возврата ходов можно рассматривать различные варианты указанного дебюта. Для контроля полученных знаний предлагается выполнить ряд заданий. В первом задании необходимо указать ходы, определяющие данный дебют. В заданиях 2 и 3 система генерирует некоторую позицию, характерную для данного дебюта, и просит сделать какой-либо из рекомендованных теорией ходов соответственно за белых или за черных. Далее, если вариант достаточно длинный, машина отвечает на указанный пользователем ход (при условии, конечно, что он соответствует одному из ходов в памяти) и снова передает очередь хода человеку.

Контрольные позиции строятся путем случайного выбора ходов при постепенном возрастании глубины вариантов. Количество и сложность вопросов зависят от результатов предыдущих этапов. Обучаемый может использовать также ограниченное число подсказок.

В заключение несколько замечаний по реализации системы. Система «Дебют» реализована на персональном компьютере IBM-XT в ВЦ АН СССР. Программы написаны на языке Бейсик. Использование данного языка связано с его развитыми графическими средствами, которые активно используются при отображении диаграмм и построении деревьев, и высокой скоростью работы компилированных программ.

Поскольку IBM BASIC предоставляет только 64К памяти, система разбита на семь функционально различных программ, которые взаимодействуют довольно сложным образом. На верхнем уровне загружаемый модуль определяется главной программой MAIN при выборе пользователем одного из пунктов основного меню. Затем программы взаимо-

УДК 681.327.0+612.76

Г. Л. Курдюмов

СИСТЕМЫ РУЧНОГО УПРАВЛЕНИЯ НЕПРЕРЫВНО МЕНЯЮЩИМИСЯ СИГНАЛАМИ

Системы ручного управления непрерывно меняющимися сигналами могут использоваться для ввода в ЭВМ графической и текстовой информации, создания зрелищных световых эффектов. Несложные технические доработки систем позволят создавать с их помощью мультфильмы, конструировать электронные музыкальные инструменты, микшерские пульты для работы звукооператоров, спортивные тренажеры и дистанционное управление роботами.

Большинство существующих в настоящее время систем ручного управления предполагает наличие отдельной клавиши или ручки для каждого регулируемого сигнала (увеличение числа таких ручек требует от оператора значительного напряжения внимания и памяти). Рассматриваемые системы многомерного управления позволяют оператору свободно перемещать в пространстве ручку управления, при этом техническое устройство регистрирует ее координаты, выдавая по ним значения выходных сигналов. С использованием одной ручки управления оператор может задавать значения до шести независимо изменяемых параметров, а при работе двумя руками максимальное их число увеличивается до двенадцати. Системы такого типа учитывают возможности человека и позволяют сделать его «общение» с компьютерной техникой более эффективным и привлекательным.

Прототипами рассматриваемых устройств являются, например, устройство ввода типа «Мышь» и «Планшет» [1], в которых используются две степени свободы движения ручки управления, что дает возможность независимо регулировать не более двух непрерывно меняющихся сигналов. Черты сходства с рассматриваемыми системами управления имеет устройство, реагирующее непосредственно на положение человеческого тела,— электронно-музыкальный инструмент «Терменвокс» [2]. Устройства такого типа не обеспечивают однозначности: одному и тому же набору значений управляемых параметров соответствуют разные положения рук оператора.

Существенно бо́льшие возможности для управления непрерывно меняющимися сигналами предоставляют оператору системы многомерного управления, обладающие следующими свойствами:

возможность одновременного управления с помощью одной ручки несколькими параметрами. Ручка управления движется как одно монолитное тело (наличие в ней дополнительных подвижных частей, например кнопок, изгибов, в данном случае не рассматривается);

диапазон перемещений ручки управления, точность регистрации ее пространственных координат, максимальные скорость и ускорение движения ручки и изменения управляемых параметров определяются возможностями человека. Система неинерционна: момент инерции ее подвижных частей и силы механического трения существенно не превышают соответствующих параметров руки;

однозначность соответствия пространственных координат ручки управления значениям управляемых параметров. Это позволяет оператору четко выполнять поставленные задачи, запоминать и совершенствовать свои действия.

Устройство, в полной мере обладающее перечисленными свойствами, можно использовать в качестве универсального.

Способы технической реализации принципа многомерного управления. Простейшая система ручного управления уровнями трех электрических сигналов посредством свободного перемещения в пространстве ручки управления описана в работе [3]. Усовершенствованный ее вариант (см. рисунок) представлен в [4]. Для регистрации трех пространственных координат конца ручки управления (1) используется одна растяжимая тяга (2), три коротких промежуточных тяги (3) и три датчика натяжения (4). Особенности этого способа— небольшие габариты устройства и отсутствие сил трения.

Существуют системы многомерного управления с помощью механических стержней и шарниров. Технические требования к таким системам оказываются довольно жесткими: малый вес подвижных частей, небольшие силы трения, отсутствие люфтов, износостойкость и устойчивость к значительным скоростям.

Силы, действующие на ручку управления в процессе ее движения, можно использовать целенаправленно. В простейшем случае это позволяет оператору убирать свою руку с ручки управления, фиксируя последнюю в любом требуемом положе-

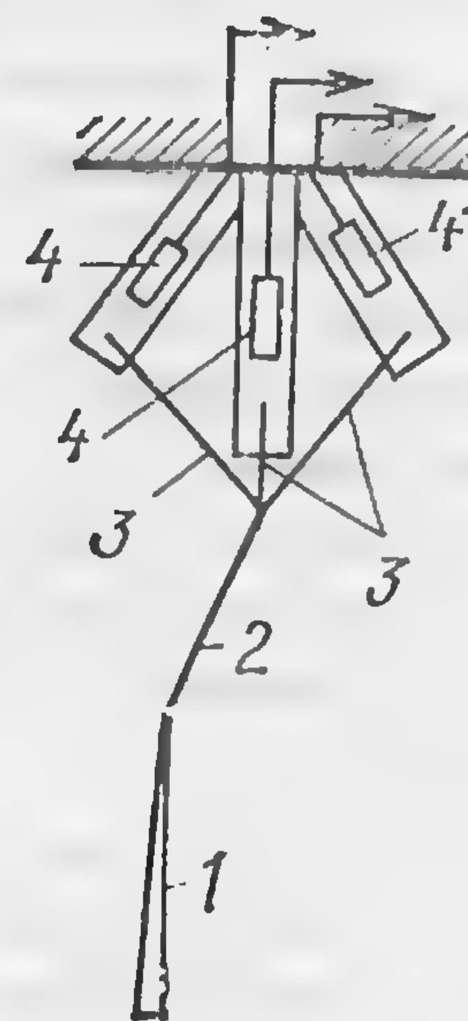


Рис. Система многомерного управления непрерывно меняющимися сигналами

действуют по указанной пользователем схеме диалога.

Некоторые характеристики системы: объем программ — 90К; среднее время выборки партии — 5 с, автоматической индексации — 12 с, поиска позиции в партии — 6 с; диск объемом 360 К вмещает около 1500 партий, винчестерский диск (10 Мбайт) — до 50 000 партий.

Определение оптимальных значений параметров и анализ статистических данных проводятся совместно с шахматистами сектора шахмат ВНИИФК.

В разработке системы принимали участие А. В. Боткин, Р. Н. Билунова, В. А. Задерман, А. А. Тимофеев.

ЛИТЕРАТУРА

1. Адельсон-Вельский Г. М., Арлазаров В. Л., Донской М. В. Программирование игр. — М.: Наука, 1978.
2. Матанович А. Информационная система в шахматах. — Шахматный бюллетень, 1983, № 11.
3. Билунова Р. И. Дебютные индексы и... мода. — Бюллетень ЦШК СССР, 1977, № 3.

Статья поступила 3 июля 1985 г.

нии, и выполнять другие функции. Возможна также передача информации в обратном направлении. Можно создать иллюзию ощупывания палочкой предмета, «протыкания» его оболочек, сообщить оператору о силах механического сопротивления, действующих на рабочие органы манипулятора.

Наибольшую свободу перемещения ручки управления обеспечивают системы многомерного управления, в которых механическая связь между ручкой управления и самим устройством полностью отсутствует. Примеры таких систем описаны в [1, 5—6].

Применение. Успешно используют системы трехмерного управления для создания зрелищных световых эффектов [7, 8]. Другой реализованный в настоящее время способ применения многомерного управления связан с вводом в ЭВМ графической информации [5, 6]. Перемещая на плоскости источник ультразвука, оператор вводит в ЭВМ информацию о траектории его движения. Это позволяет, во-первых, вводить в ЭВМ графическую информацию, уже имеющуюся, например, на бумаге, во-вторых, создавать эту информацию в процессе творческой деятельности оператора. Второе из перечисленных направлений применения представляется наиболее перспективным. Оно охватывает как ввод в ЭВМ дополнительной информации, относящейся к уже имеющимся изображениям (используется в настоящее время при анализе медицинских рентгенограмм), так и создание новых изображений. Имеется принципиальная возможность «рисовать» контуры тел сразу в трех измерениях, наблюдая их на экране дисплея в любом требуемом ракурсе.

Трехмерное управление цветом, основанное на соответствии каждому оттенку некоторой точки пространства [8], позволяет эффективно работать с полутоновым дисплеем. Для введения в память ЭВМ информации о распределении цветовых оттенков в плоскости экрана оператору можно предоставить широкий выбор «инструментов», в том числе таких, которые имитируют традиционные инструменты художников и дизайнеров: различные кисти, пульверизатор и т. п. Некоторые из этих возможностей в настоящее время частично реализованы с помощью устройства «Мышь». Но тот факт, что при его перемещении регистрируются только две пространственные координаты, ограничивает возможности применения. Проведем аналогию: художники уделяют немало внимания качеству и состоянию их инструментов. Форма и вес кисти, оттенки красок и условия их смещения на палитре — все имеет значение и становится составной частью творческого процесса. Трехмерная ручка управления в сочетании с компьютером и дисплеем позволяет развивать новое направление изобразительного искусства — электронную живопись. Программными средствами можно придавать получаемым изображениям различные виды движения и деформации, что позволит использовать их для создания мультфильмов. Полутоновые дисплеи можно использовать также для выдачи оператору большого объема образно-упорядоченной визуальной информации в системах управления разного функционального назначения. В этом случае многомерное управление позволит вводить в интерфейс машины достаточное количество исходных образов, выражаемых формой и цветом предъявляемых объектов.

Известно, какой популярностью пользуются сейчас компьютерные игры, с помощью многомерного управления они становятся еще и подвижными, движение предметов может стать в них объектом моделирования, потребовав от игрока точных, координированных движений.

Системы многомерного управления можно использовать взамен традиционной клавиатуры для ручного ввода в ЭВМ текстовой или цифровой информации. Потребуется установить подходящую *знаковую систему*, то есть соответствие между вводимыми в машину символами и участками траектории движения ручки управления. Один из вариантов — имитация клавиатуры. Используя трехмерную ручку управления [4], можно за счет изменения двух ее координат перемещать курсор по экрану дисплея, совмещая его с видимыми изображениями клавиш. Одному фиксированному (но не крайнему) значению третьей координаты ручки соответствует «поверхность ввода»: в те моменты, когда конец ручки пересекает эту «поверхность», знак соответствующей клавиши считается введенным и отображается на экране отдельно от изображения клавиатуры. Работу с такой системой можно сравнить с печатью на клавиатуре одним пальцем; но возможность проходить воображаемую клавиатуру «насквозь» позволяет уменьшить вдвое число поворотов в движении руки, необходимое для ввода того или иного текста, что повышает скорость работы. По мере приобретения оператором соответствующего навыка изображение клавиш на экране окажется ненужным. Использование воображаемой клавиатуры обеспечивает до некоторой степени преемственность навыков работы по отношению к традиционному устройству ввода.

Другой вариант знаковой системы сопоставляет знаки текста с направлениями движения конца ручки. Чтобы вводить подряд несколько одинаковых знаков, а также возвращать ручку в удобное положение в случае преобладания в тексте тех или иных знаков, соответствующих близким направлениям, часть направлений считается пустыми, то есть не соответствующими определенному знаку. Таким образом, для ввода повторяющихся знаков можно будет чередовать движение в направлении, соответствующем этому знаку, с движением в каком-либо пустом направлении. Последовательно перемещая ручку по различным пустым направлениям, можно вернуть ее в любое удобное положение. В знаковой системе такого рода длина участков траектории, соответствующих отдельным знакам, может быть совсем небольшой; средняя величина углов поворота траектории меньше, чем при знаковой системе, моделирующей клавиатуру. Каждому вводимому слову соответствует определенный зигзаг, последовательность смены направлений движения. Часто повторяемые зигзаги со временем хорошо запомнятся, и их можно будеть легко повторять без напряжения внимания. Это можно сравнить с трехмерной стенографией. Окончательный вывод о целесообразности использования той или иной знаковой системы может быть получен в результате экспериментов.

В связи со всем сказанным при конструировании средств вычислительной техники представляется в ряде случаев возможным отказаться от использования традиционной клавиатуры, уменьшив тем самым габариты устройств. Открывается перспектива создания переносной персональной ЭВМ с широкими возможностями обмена информацией в диалоговом режиме.

ЛИТЕРАТУРА

1. У. Ньюмен, Р. Спрулл. Основы интерактивной машинной графики.— М.: Мир, 1976, с. 182—195.
2. Л. С. Термен. Физика и музыкальное искусство.— М.: Знание, 1966.
3. Г. Л. Курдюмов, В. С. Смолин. Устройство управления световыми эффектами / А. с. № 1026821, СССР БИ № 25, 1985.
4. Г. Л. Курдюмов. Координатное устройство управления цветом / А. с. № 1118387, СССР БИ № 38, 1984.
5. Э. К. Скворцов. Устройство для считывания графической информации / А. с. № 377584, СССР БИ № 40, 1981.
6. Н. Лохов. Приоритет огромного значения.— Изобретатель и рационализатор, 1984, № 8, с. 6—8.
7. С. Волков. Пространство света.— Изобретатель и рационализатор, 1983, № 8, с. 19.
8. Г. Л. Курдюмов. Трехмерный цвет.— Изобретатель и рационализатор, 1985, № 2, с. 24.

Статья поступила 5 апреля 1985 г.

УДК 681.322.1

В. Н. Барышников, В. П. Быстров, М. А. Воронов,
Ф. И. Паначев, В. Ю. Романов

МОДУЛЬ ПРОЦЕССОРА ПЕРСОНАЛЬНОЙ ЭВМ «ИРИША»

(Продолжение цикла статей по компьютеру «Ириша». Начало см. в «МП» № 3, 1985 г. и № 1, 1986 г.)

Модуль процессора является одним из двух основных модулей ПЭВМ «ИРИША». Он не имеет собственного ОЗУ и устройств для отображения информации, поэтому для его работы необходим как минимум модуль телевизионного адаптера *. Взаимодействие модулей осуществляется через системную магистраль В модуле процессора находятся собственно процессор, средства для обмена с системной магистралью, ПЗУ емкостью 4...16К байт, интерфейс клавиатуры, последовательный интерфейс для работы с магнитофоном и локальной сетью, параллельный интерфейс принтера, таймер, синтезатор звука и двухканальный игровой адаптер (рис. 1).

Модуль процессора имеет свою локальную магистраль, через которую происходит обмен информацией между отдельными его узлами. Локальная магистраль отделена от системной набором буферных усилителей с тремя состояниями на выходе, что позволяет при необходимости отключать модуль от системной магистрали. Сигналы локальной магистрали, а также ряд внутренних сигналов управления и синхронизации выведены на специальный разъем, к которому подключается технологическое оборудование при проведении наладочных и ремонтных работ. При необходимости к этому же разъему могут подключаться дополнительные платы, расширяющие возможности основной конфигурации модуля. Конструктивно такие платы располагаются над основной платой. Для некоторых исполнений «Ириши» в этот разъем устанавливается дополнительное ПЗУ емкостью 2К байт с тем, чтобы общий объем памяти был равен 16К байт. Кроме того, на плате процессора расположены буферное ОЗУ контроллера накопителя на гибких дисках и схема ускоренной пересылки данных.

Процессор

Основу узла процессора составляет микропроцессор (МП) КР580ВМ80А со схемами обрамления КР580ГФ24 (тактовый генератор) и КР580ВК28 (системный контроллер). МП работает с тактовой частотой 1.7777 МГц, получаемой из опорной частоты ПЭВМ 16 МГц. Сигналы опорной и тактовой частот процессора используются другими модулями, подключаемыми к системной магистрали. В частности, частота 16 МГц необходима для работы модулей телевизионного адаптера и контроллера накопителя на гибких магнитных дисках.

Узел процессора, кроме сигналов синхронизации, вырабатываемых системным контроллером, формирует дополнительные сигналы выборки памяти или устройств ввода-вывода $IO/\overline{M}$, чтения-записи $R/\overline{W}$, строба обмена $\overline{TE}$ Последние два сигнала необходимы для работы с системной магистралью. Кроме того, в состав узла входит схема формирования сигнала требования захвата магистрали HLD. Для синхронизации обмена информацией между узлом процессора, внешними устройствами и памятью используется сигнал готовности $\overline{READY}$. В отсутствие этого сигнала МП ожидает завершения операции обмена. Сигнал готовности используется для организации режима асинхронного обмена по системной магистрали и для пошагового исполнения команд во время наладки и ремонта. Внутренние

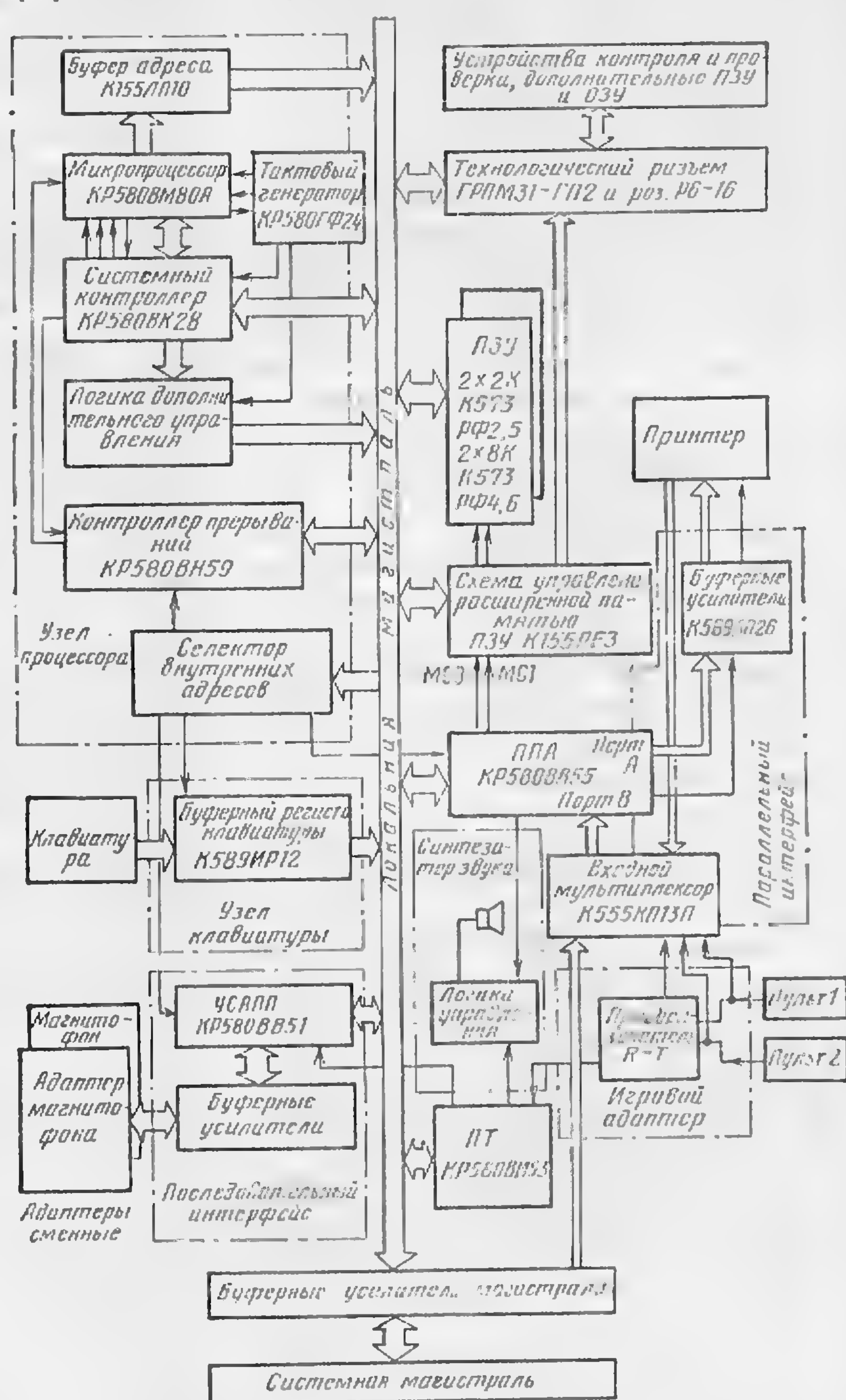


Рис. 1. Структурная схема модуля процессора

* Романов В. Ю., Барышников В. Н., Воронов М. А., Паначев Ф. И. **Графические возможности персональной ЭВМ «Ириша».**— Микропроцессорные средства и системы, 1986, № 1, с. 61—72.

устройства модуля, включая ПЗУ, работают с МП синхронно без использования сигнала готовности. Это позволяет в значительной степени компенсировать неудобства, возникающие из-за неопределенного времени исполнения команд при асинхронном способе обмена.

Модуль процессора позволяет работать с устройствами внутренней и внешней памяти, общий объем которой превышает 64К байт, непосредственно адресуемой МП. Для управления расширенной памятью в состав внутренних устройств модуля включен двухбитный регистр, организованный как часть порта С БИС программируемого периферийного адаптера (ППА) КР580ВВ55 (рис. 2). Выходные сигналы этого регистра и три старших разряда адресной шины МП поступают на входы ПЗУ преобразователя адресов. Выходная информация ПЗУ используется для формиро-

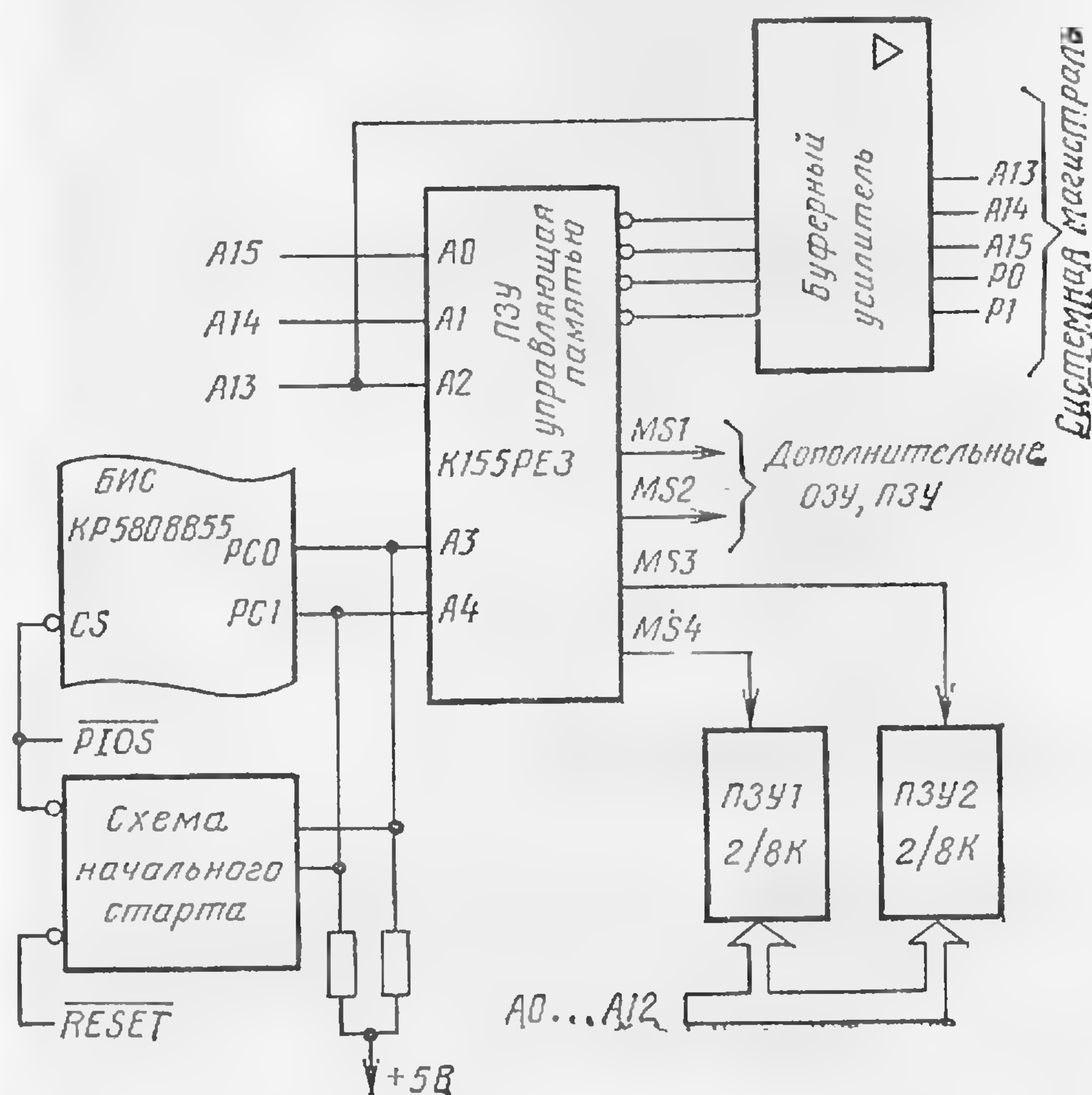


Рис. 2. Функциональная схема узла управления расширенной памятью

вания 18-разрядного адреса в системной магистрали и для управления включением внутренних ПЗУ. Использование ПЗУ типа К155РЕ3 позволяет иметь четыре различные карты распределения памяти при минимальном размере сегмента 16К байт. Минимальный размер сегмента внутреннего ПЗУ 8К байт. На физическом уровне такой узел позволяет адресовать до 256К байт памяти, однако реально из-за необходимости организации областей связи между частями программы этот объем несколько меньше. Кодировка ПЗУ управления памятью определяется требованиями программного обеспечения. Указанное ПЗУ является съемным (микросхема монтируется в панельку).

При начальном старте, а также при получении сигнала $\overline{\text{RESET}}$ с помощью специальной схемы начального пуска включается нулевая карта распределения (сигналы МС0 и МС1 устанавливаются низким уровнем). Общее требование к ПЗУ управления памятью состоит в том, чтобы в начальных адресах памяти находилось внутреннее ПЗУ модуля, содержащее стартовый монитор системы.

Модуль процессора, как уже отмечалось, имеет возможность отключаться от системной магистрали ПЭВМ. Освобождение магистрали производится по получению сигнала требования на захват $\overline{\text{BUSRQ}}$ двумя способами. Во-первых, освобождение осуществляется на аппаратном уровне без вмешательства программы с помощью генерации сигнала требования на захват магистрали HLD. Такой способ эффективен, когда устройство, запрашивающее магистраль, занимает ее на короткое время. Когда освобождение требуется на значительный промежуток времени и необходимо, чтобы модуль процессора продолжал работу со своими аппаратными средствами (например, реагировал на коды, получаемые от клавиатуры), полезным оказывается отключение по команде установки на выходе PC4 порта С БИС ППА «Лог. 1». Сигнал $\overline{\text{BUSRQ}}$ в таком режиме не влияет на работу МП, но он может быть проанализирован программно.

При выполнении операции Сброс за счет того, что все порты ППА переводятся в режим ввода информации, на выходе PC4 устанавливается высокий уровень, что, в свою очередь, приводит к отключению модуля от системной магистрали. Таким образом, при старте модуль работает автономно до момента передачи соответствующей команды в БИС ППА.

В модуле процессора реализована 8-уровневая система прерываний. Аппаратно она базируется на БИС контроллера прерываний (КП) КР580ВН59 (рис. 3).

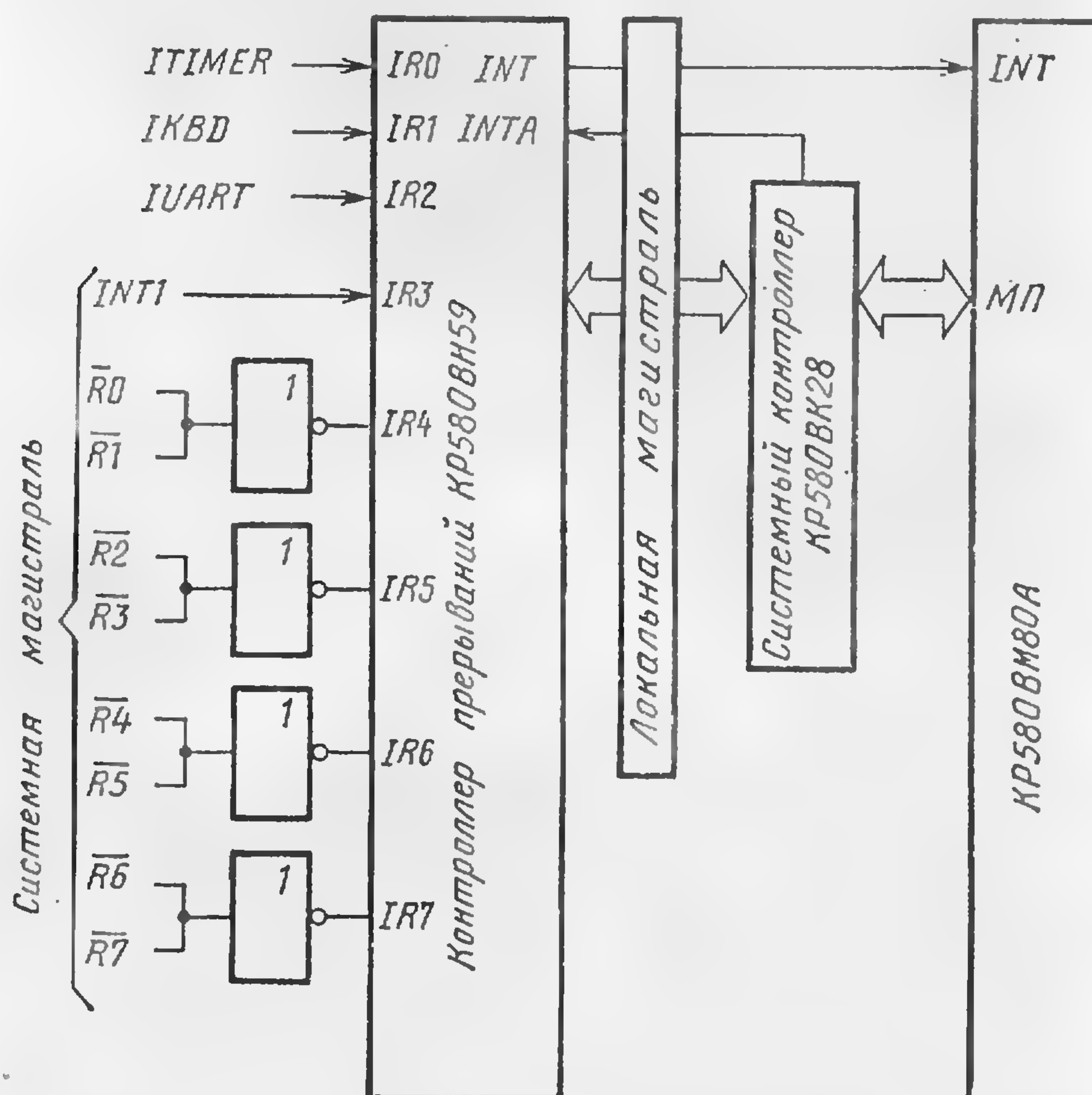


Рис. 3. Схема узла прерываний

Из восьми входов запроса прерываний три с наиболее высоким приоритетом используются самим модулем, а остальные пять поступают с соответствующих линий системной магистрали. Наивысшим приоритетом обладает сигнал прерывания ITIMER, вырабатываемый БИС программируемого таймера (ПТ) КР580ВИ53. Он используется для организации работы программ, требующих привязки к реальному масштабу времени. Сигналы прерывания IKBD и IUART сообщают процессору о том, что от клавиатуры или из локальной сети принят байт информации и требуется его обработка.

Внутренние периферийные узлы

Периферийные узлы, расположенные на плате модуля процессора, обеспечивают функционирование ПЭВМ в минимальном комплекте. Большинство узлов выпол-

нено на базе БИС микропроцессорного комплекта серии КР580. В табл. 1 приведены адреса внутренних устройств модуля с кратким описанием их назначения.

Описание работы каждого узла полезно начать с функций БИС ППА КР580ВВ55, поскольку эта микросхема используется в работе многих из них. БИС ППА состоит из трех регистров-портов, которые могут работать как на ввод, так и на вывод информации. В данной конкретной реализации порт А может использоваться как для ввода, так и для вывода данных, порт В — только для ввода, а порт С — только для вывода. Выходы порта С используются для управления работой внутренних схем модуля. ППА имеет возможность изменять состояние отдельных выводов порта С при помощи специальных команд, передаваемых в регистр управления. На входе порта В для расширения возможностей по объему вводимой информации установлен мультиплексор на основе ИС К555КП13, одновременно выполняющий функции входного буфера и защелки на время ввода. Управление мультиплексором осуществляется выходным сигналом РС7 порта С ППА. Назначение и функции входов и выходов ППА приведены в табл. 2.

Интерфейс клавиатуры

Алфавитно-цифровая клавиатура, подключаемая к модулю процессора через параллельный байтный регистр, при нажатии на клавишу должна вырабатывать код символа, согласно кодировке КОИ-8, стробируемый сигналом $\overline{STB}$ (активный низкий уровень) (рис. 4). По фронту сигнала $\overline{STB}$ взводится внутренний триггер прерывания приемного регистра, выполненного на ИС 5589ИР12, устанавливается сигнал готовности и прерывания IKBD с помощью которого указывается, что во входном регистре находится код принятого символа. Таким образом, заполнение приемного регистра контролируется с помощью анализа сигнала прерывания или седьмого бита байта состояния, считываемого из порта В ППА (канал В входного мультиплексора).

Рис. 4. Интерфейс клавиатуры (*а*), временные диаграммы обмена с клавиатурой (*б*)

Таблица 1

Адреса внутренних узлов модуля процессора ПЭВМ «ИРИША»

Адрес Hex	Вид операции	Назначение	Примечание
00 ... 03	—	Резервные адреса устройств расширения модуля	В минимальных комплектах не используется
04 05	R R	Регистр данных клавиатуры Регистр данных клавиатуры (дублирует адрес 04Н)	ИС К589ИР12
06 07	R/$\overline{W}$ R/$\overline{W}$	Регистр данных последовательного интерфейса Регистр управления статуса последовательного интерфейса	БИС КР580ВВ51
08 09 0A 0B	R/$\overline{W}$ R/$\overline{W}$ R/$\overline{W}$ $\overline{W}$	Считывание / загрузка счетчика 0 Считывание / загрузка счетчика 1 Считывание / загрузка счетчика 2 Запись байтов управления	БИС КР580ВИ53
0C 0D 0E 0F	R/$\overline{W}$ R/$\overline{W}$ R/$\overline{W}$ R/$\overline{W}$	Считывание состояния регистра запроса прерывания и обслуживания запроса. Запись команд ICW1, OCW2, OCW3 Считывание регистра маски прерываний, запись команд ICW2, ICW3, ICW4, OCW1 Дублирует адрес OCH Дублирует адрес ODH	БИС КР580ВН59
10 12 11 13	R/$\overline{W}$ R/$\overline{W}$ R/$\overline{W}$ $\overline{W}$	Считывание / запись информации, порт А Считывание / запись информации, порт В Считывание / запись информации, порт С Регистр управления БИС ППА	БИС КР580ВВ55
14 ... 1В	—	Резервные адреса дополнительных устройств модуля	В минимальных комплектах не используются

Таблица 2

Функции входов и выходов БИС ППА КР580ВВ55

Сигнал	Назначение	
	PC7 = 0	PC7 = 1
PA0	Вход-выход ПИ 0	Старт цикла преобразователя игрового адаптера
PA1	Вход-выход ПИ 1	Выбор игрового пульта. Запись по сигналу РС6. «0» — пульт 1, «1» — пульт 2
PA2	Вход-выход ПИ 2	Не используется
PA3	Вход-выход ПИ 3	
PA4	Вход-выход ПИ 4	Канал побитного управления звуком (шумовой канал)
PA5	Вход-выход ПИ 5	Разрешение работы канала 1 синтезатора звука. «0» — разрешена
PA6	Вход-выход ПИ 6	Не используется
PA7	Вход-выход ПИ 7	Выход программного канала последовательного интерфейса
PB0	Вход статуса ПИ S13	Анализ завершения цикла преобразования игрового адаптера: «0» — завершен
PB1	Вход статуса ПИ S14	Анализ состояния кнопки игрового пульта 1 : «0» — нажата
PB2	Вход статуса ПИ S15	Анализ состояния кнопки пульта 2 : «0» — нажата
PB3	Вход статуса ПИ S16	Вход программного канала последовательного интерфейса
PB4	Вход статуса ПИ S12	Анализ сигнала BUSRQ системной магистрали: «1» — установлен
PB5	Вход статуса ПИ S11	Анализ сигнала PON системной магистрали: «1» — норма
PB6	Вход анализа типа дополнительной платы EX1	Анализ сигнала INT1 системной магистрали: «1» — установлен
PB7	Вход анализа типа дополнительной платы EX2	Анализ готовности клавиатуры: «1» — код принят
PC0	Выход управления ПИ SO1	
PC1	Выход управления ПИ SO2	
PC2	Управление выбором рабочей карты памяти MC0	
PC3	Управление выбором рабочей карты памяти MC1	
PC4	Разрешение захвата системной магистрали: «1» — захват разрешен	
PC5	Выбор режима игровой адаптер — синтезатор звука: «1» — игровой адаптер, «0» — синтезатор звука	
PC6	Строб записи в дополнительный регистр управления игровым адаптером и синтезатором звука	
PC7	Выбор функций портов А и В	

После считывания из регистра принятого символа сигнал IKBD сбрасывается и схема готова к приему нового кода. Кроме того, узел вырабатывает сигнал IBF, который передается назад в клавиатуру и указывает ей, что входной буфер заполнен и новый символ принят быть не может. После считывания кода из приемного регистра снимается также сигнал $\overline{IBF}$. Кроме указанных сигналов, клавиатура должна формировать потенциальный сигнал $\overline{RESET}$ (активный низкий уровень), из которого вырабатывается сигнал Сброс МП. Интерфейс допускает подключение серийной клавиатуры 15ВВВ-97-006 без каких-либо переделок. Вместо нее может использоваться любая другая клавиатура, имеющая аналогичный протокол обмена.

Последовательный интерфейс

Последовательный интерфейс (рис. 5) предназначен для организации межмашинной связи, подключения стандартных дисплейных устройств, работы с бытовым магнитофоном, используемым для записи программ и данных. Интерфейс имеет два канала обмена. Канал 1 построен на базе БИС КР580ВВ51. Канал 2 реализуется программно: по выводу за счет манипуляций с выходом РА7 порта А ППА, а по вводу обработкой данных, получаемых с входа РВ3 порта В той же БИС. Сигналы каналов 1 и 2 после соответствующей буферизации выводятся на общий разъем последовательного интерфейса. Входные и выходные сигналы на этом разъеме соответствуют стандартным ТТЛ-уровням. Между выходом последовательного интерфейса модуля и конкретным используемым устройством должен быть включен промежуточный адаптер. Обычно такой адаптер состоит из двух-трех микросхем буферных усилителей. Например, адаптер для работы с дисплеем 15ИЭ-00-13 состоит из двух элементов оптической развязки, и для реализации выхода на стандартный интерфейс ИРПС требуются только две микросхемы преобразователей уровня. При организации локальной сети схема адаптера монтируется в розетках для соединения машин. Использование вынесенных из модуля промежуточных адаптеров позволяет учитывать специфику работы подключаемых к этому каналу устройств.

Скорость обмена в канале 1 задается путем программирования БИС КР580ВВ51 и канала 0 БИС КР580ВИ53. Этот канал используется в качестве программируемого делителя частоты и из тактовой частоты Ф2TTL, равной 1.7777 МГц, вырабатывает необходимую частоту для каналов приемника и передатчика БИС КР580ВВ51. Для задания максимальной скорости обмена, равной 9600 бод, при программировании БИС КР580ВВ51 в режим с входной частотой синхронизации приемника и передатчика, в 16 раз большей скорости обмена, указанный канал ПТ должен работать делителем частоты на 11.57. Поскольку дробные коэффициенты деления задать невозможно, он устанавливается равным 12. Это приводит к отклонению скорости обмена относительно стандартной на 3,7 %, что допустимо в асинхронном режиме работы. При более низких скоростях обмена это отклонение меньше.

Канал 2 менее скоростной. Реально он может поддерживать скорости обмена до 1200 бод. Поскольку все временные соотношения задаются программно, время исполнения команд должно быть строго определенным. Выполнение этого требования возможно только в том случае, если программа записана в системном ПЗУ модуля. Канал 2 используется для работы с бытовым магнитофоном. Специальная программа, содержащаяся в системном ПЗУ, позволяет записывать и считывать программы, текстовые файлы и блоки данных. (Программа с комментариями будет приведена в следующем выпуске «Учебного центра».) Метод и формат записи соответствуют принятому в стандарте MSX. Скорость обмена при работе с бытовым магнитофоном 1200 бод. Для работы с магнитофоном также необходим промежуточный адаптер, использующий микрофонный вход и линейный выход магнитофона.

Рис. 5. Последовательный интерфейс

Кроме того, желательно, чтобы магнитофон имел вход для дистанционного отключения двигателя. Если такого входа нет, то работа с текстовыми файлами или файлами данных будет невозможна. Программа работы с магнитофоном использует для вывода выход PA7 порта A ППА, для включения двигателя магнитофона выход DTR БИС последовательного интерфейса, а для ввода информации вход DSR той же БИС. Схема адаптера магнитофона приведена на рис. 6.

Параллельный интерфейс

Параллельный интерфейс (рис 7) предназначен для подключения принтера, но, кроме того, он может выполнять универсальные функции и использоваться для работы с другими устройствами.

Рис. 6. Адаптер кассетного магнитофона

Информационные линии интерфейса $\overline{P0}...\overline{P7}$ — двунаправленные, причем направление передачи определяется состоянием входа У. Кроме того, в его состав входят линии $\overline{SO1}$ и $\overline{SO2}$, работающие только на вывод, и 6 линий SI1...SI6, работающих только на ввод информации. При работе с принтером 15УВВП-30-004 специальная драйверная программа реализует стандартный интерфейс ИРПР. В этом случае назначение линий следующее: $\overline{PO}...\overline{P7}$ — информационные линии, $\overline{SO1}$ — сигнал готовности ГИ, $\overline{SO2}$ — стробирующий сигнал СТР, SI1 — вход запроса ЗП, SI2 — подтверждение приема ГП, сигналы SI3...SI6 соответствуют сигналам статуса СП0...СП3. Вход управления направлением передачи должен быть на разъеме принтера соединен с общим выводом.

В схемотехническом отношении основу интерфейса составляет часть БИС ППА КР580ВВ55. Порт A этой микросхемы в зависимости от режима может быть использован для ввода или вывода информации. Генерация сигналов управления осуществляется с помощью выходов PC0 и PC1 порта C ППА. Поскольку шесть остальных выходов используются для управления другими узлами модуля, то для изменения состояния выходов PC0 и PC1 рекомендуется использовать команды раздельного управления выходами, передаваемыми в регистр управления ППА.

Рис. 7. Параллельный интерфейс

Таймер

Канал 1 БИС КР580ВИ53 используется в качестве программно управляемого генератора временных интервалов. Опорная частота формируется делением частоты сигнала Ф2TTL на 256. Такая опорная частота позволяет задавать временные интервалы в диапазоне длительности от 432 мкс до 9,4 сек. с дискретностью 144 мкс. Выходной сигнал канала поступает на один из входов запроса прерываний БИС КП КР580ВИ59 и имеет наивысший приоритет. При программировании режима работы указанного канала ПТ следует учиты-

вать, что БИС контроллера прерываний фиксирует поступающие запросы по фронту сигнала, но не защелкивает их. Поэтому если время нахождения сигнала запроса в активном состоянии недостаточно для его программной обработки, то может возникнуть ошибочная ситуация, при которой устанавливается сигнал прерывания МП, а его источник не фиксируется. Для исключения подобных явлений рекомендуется устанавливать длительность выходных импульсов запроса прерывания больше времени, необходимого для программной обработки. Кроме того, при составлении программ необходимо учитывать, что некоторые системные программы блокируют прерывания на время около 0,4 с.

Синтезатор звука

Синтезатор (рис. 8) предназначен для формирования программно управляемых звуковых сигналов. Он включает в себя два независимых канала для генерации

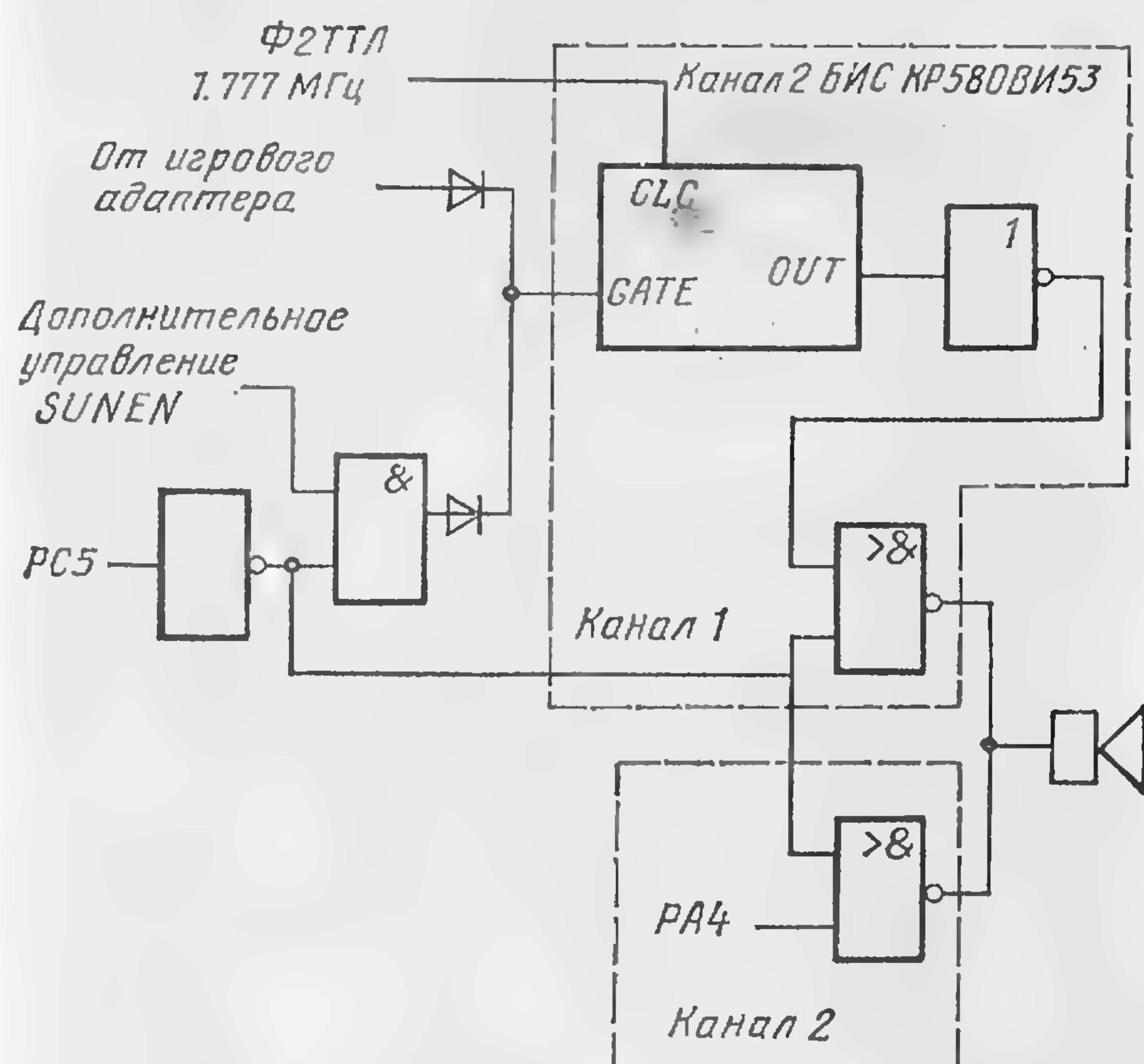


Рис. 8. Синтезатор звука

сигналов звуковой частоты К а н а л 1 вырабатывает звуки заданной частоты и музыкальные заставки. Он выполнен на базе одного из каналов БИС КР580ВИ53, используемого в качестве программируемого делителя частоты. Задавая коэффициенты деления, можно изменять частоту вырабатываемого сигнала (табл 3). Управление включением и выключением канала 1 производится по программе специальным триггером, входящим в состав регистра дополнительного управления

К а н а л 2 предназначен для синтеза шумовых сигналов. Сигнал на выходе задается последовательностью байтов, записываемых в регистр порта А БИС КР580ВВ55. Управление состоянием выхода этого канала производится четвертым битом (D4) указанного байта Изменяя время между передачами, можно соответствующим образом формировать выходной сигнал (рис. 9). Следует отметить, что работой обоих каналов управляет выход PC5 порта С ППА. Звуковые сигналы вырабатываются в том случае, если на этом выходе установлен низкий уровень («Лог. 0»).

В качестве излучателя звука использован малогабаритный динамический громкоговоритель 0,25ГД10, смонтированный над основной платой модуля и крепящийся к ее арматуре. Усилителем для такого излучателя служит часть микросхемы К155ЛА13.

Таблица 3

Коэффициенты деления для синтеза нот

Октава	Нота	Коэффициент деления	Старший байт	Младший байт
Большая	МИ	10 787	2AH	23H
	ФА	10 181	27H	C5H
	ФА♯	9 610	25H	8AH
	СОЛЬ	9 070	23H	6EH
	СОЛЬ♯	8 561	21H	71H
	ЛЯ	8 081	1FH	91H
	ЛЯ♯	7 627	1DH	CBH
	СИ	7 199	1CH	1FH
Первая	ДО	6 795	1AH	8BH
	ДО♯	6 670	1AH	0EH
	РЕ	6 054	17H	A6H
	РЕ♯	5 714	16H	52H
	МИ	5 393	15H	11H
	ФА	5 091	13H	E3H
	ФА♯	4 805	12H	C5H
	СОЛЬ	4 535	11H	B7H
	СОЛЬ♯	4 281	10H	B9H
	ЛЯ	4 040	0FH	C8H
	ЛЯ♯	3 814	0EH	E6H
	СИ	3 600	0EH	10H
Вторая	ДО	3 398	0DH	46H
	ДО♯	3 207	0CH	87H
	РЕ	3 027	0BH	D3H
	РЕ♯	2 857	0BH	29H
	МИ	2 697	0AH	89H
	ФА	2 545	09H	F1H
	ФА♯	2 402	09H	62H
	СОЛЬ	2 268	08H	DCH
	СОЛЬ♯	2 140	08H	5CH
	ЛЯ	2 020	07H	E4H
	ЛЯ♯	1 907	07H	73H
	СИ	1 800	07H	08H
Третья	ДО	1 699	06H	A3H
	ДО♯	1 603	06H	43H
	РЕ	1 513	05H	E9H
	РЕ♯	1 428	05H	94H
	МИ	1 348	05H	44H
	ФА	1 273	04H	F9H
	ФА♯	1 201	04H	B1H
	СОЛЬ	1 134	04H	6EH

Игровой адаптер

Игровой адаптер (рис 10) обеспечивает подключение двух пультов, необходимых для работы с различными игровыми программами. Каждый пульт включает переменный резистор с максимальным сопротивлением 47 кОм и кнопку, смонтированные в одном корпусе. Вместо двух пультов можно подключить джойстик. Адаптер представляет собой двухканальный преобразователь сопротивление — код со схемой анализа кода преобразования и состояния кнопок пультов.

Преобразователь выполнен на базе интегрального таймера КР1006ВИ1 и канала 2 БИС КР580ВИ53 (тот же канал используется в синтезаторе звука). Процесс преобразования величины сопротивления в код происходит при участии процессора и выполняется с помощью специальной подпрограммы. Под ее контролем на выходе PA0 порта А ППА вырабатывается импульс запуска одновибратора на основе интег-

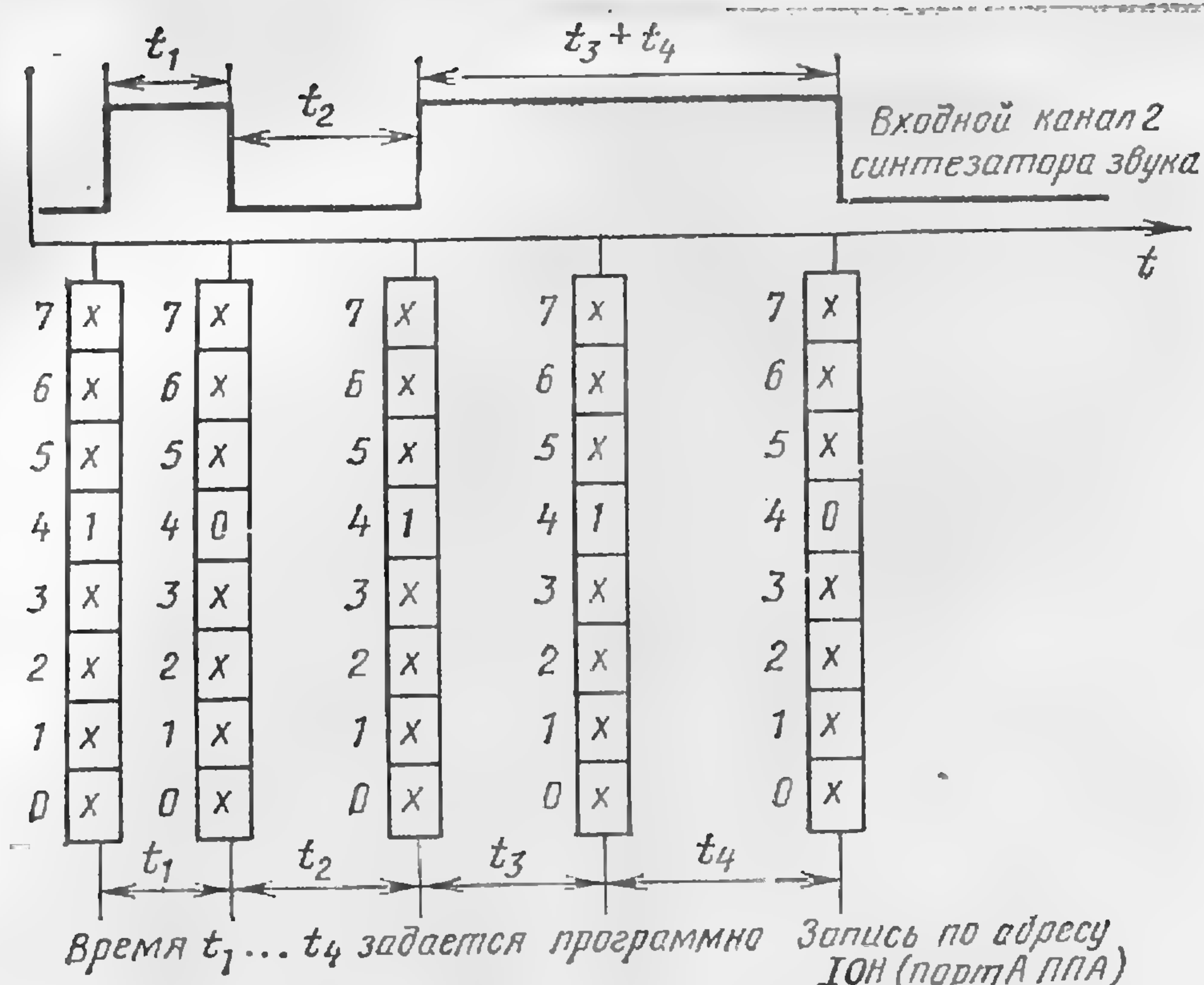


Рис. 9. Схема работы канала 2 синтезатора звука

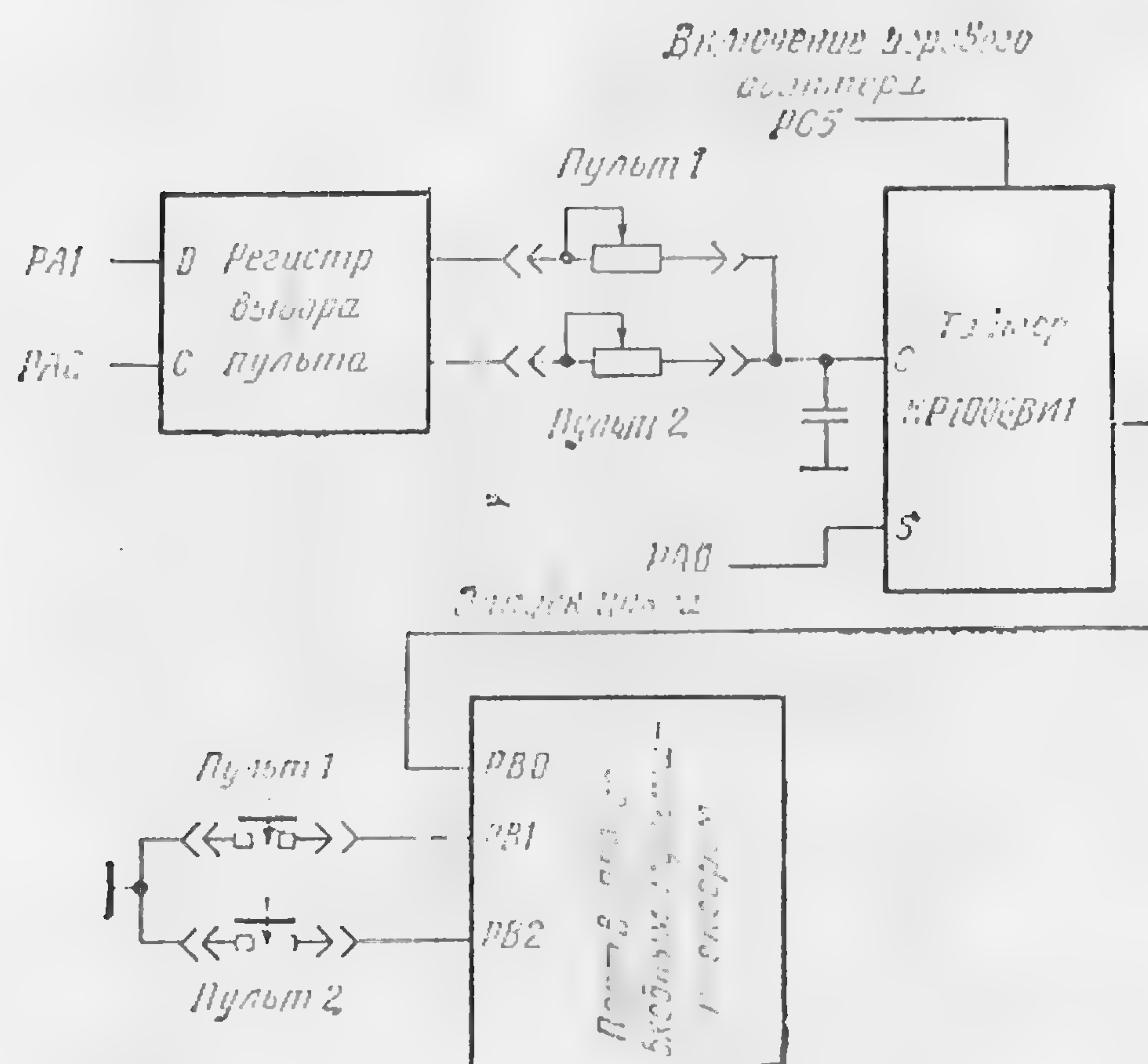


Рис. 10. Игровой адаптер

рального таймера. Одновибратор формирует на выходе импульс, длительность которого зависит от величины сопротивления переменного резистора игрового пульта и измеряется счетчиком канала 2 ПТ КР580ВИ53. Дискретность отсчета определяется тактовой частотой, поступающей на соответствующий вход ПТ, и составляет в данном случае 0,5625 мкс. Поскольку счетчик в ПТ может работать только в режиме обратного счета, процессор первоначально должен занести в него максимальное значение, равное FFFFH. Уменьшение содержимого счетчика является линейной функцией от длительности импульса, формируемого одновибратором. Считанное из счетчика значение МП преобразует в величину, пропорциональную углу поворота переменного резистора пульта. Время преобразования зависит от текущей величины резистора, но не превышает 36 мс.

Переключение переменных резисторов пультов производится с помощью дополнительного регистра, выполненного на основе КМОП микросхемы К176ТМ1. Запись в него производится с помощью подпрограммы, формирующей необходимую последовательность информационных и стробирующих сигналов. Переменные резисторы пультов подключены к выходам этого регистра. Завершение цикла преобразования контролируется процессором путем считывания байта состояния из порта В ППА.

Принципиальная электрическая схема модуля процессора приведена на развороте, монтажная схема представлена на рис. 11, топология печатной платы — на рис. 12, а, б.

Внутреннее ПЗУ

Модуль процессора имеет две розетки для установки микросхем ПЗУ. Конструктивно они выполнены таким образом, что позволяют монтировать микросхемы различной информационной емкости. Минимальный комплект состоит из двух микросхем К573РФ5 емкостью по 2К байт каждая. Дальнейшее наращивание объема производится заменой на микросхемы К573РФ41—44, К573РФ4 или К573РФ6.

В зависимости от объема ПЗУ изменяется набор записанных в них программ. В комплект ПЗУ минимального объема входят программа холодного старта и начального диалога, программы работы с кассетным магнитофоном, дисковый загрузчик, программы вывода символов на экран со знакогенератором (консольный вывод), приема символов от клавиатуры (консольный ввод), а также ряд вспомогательных программ. (Основные из перечисленных программ будут приведены в следующем выпуске «Учебного центра».) Все остальные программы должны загружаться в ОЗУ с магнитофона или диска. В дисковых вариантах при работе с малым объемом программ, хранимых в ПЗУ, особых неудобств не возникает. При работе с магнитофоном, безусловно, более удобно иметь в постоянной памяти максимальный набор программ, которым предполагается пользоваться, с тем, чтобы не тратить время на их загрузку. Окончательный набор программ для «прошивки» в ПЗУ пока не определен. Но в первую очередь в него войдут программы расширенного консольного вывода, графические подпрограммы, подпрограммы звукового синтеза и ряд дополнительных служебных подпрограмм, улучшающих эксплуатационные характеристики ПЭВМ.

Интерпретатор языка Бейсик, соответствующий MSX стандарту, требует ПЗУ емкостью 35К байт, поэтому пока он загружается с магнитной ленты или диска. В настоящее время можно занести в ПЗУ вариант языка Форт, занимающий около 5К байт. Существенно, что модуль процессора допускает расширение объема ПЗУ не только путем замены одних микросхем на другие, но и благодаря увеличению их числа. Дополнительные ПЗУ устанавливаются на специальной плате, подключаемой к технологическому разъему. Такое расширение возможно и при использовании микросхем большей информационной емкости.

Наладка модуля

Наладка модуля процессора в отличие от других модулей ПЭВМ не может производиться через системную магистраль, поэтому для контроля и проверки используется специальное оборудование: наладочный пульт, индикатор состояния линий системной магистрали и набор тестовых ПЗУ.

Н а л а д о ч н ы й   п у л ь т представляет собой устройство, с помощью которого можно наблюдать состояние сигналов на линиях локальной магистрали. Индикация осуществляется на выносном светодиодном табло в позиционном виде и в виде шестнадцатерич-

Рис. 11. Монтажная схема модуля процессора

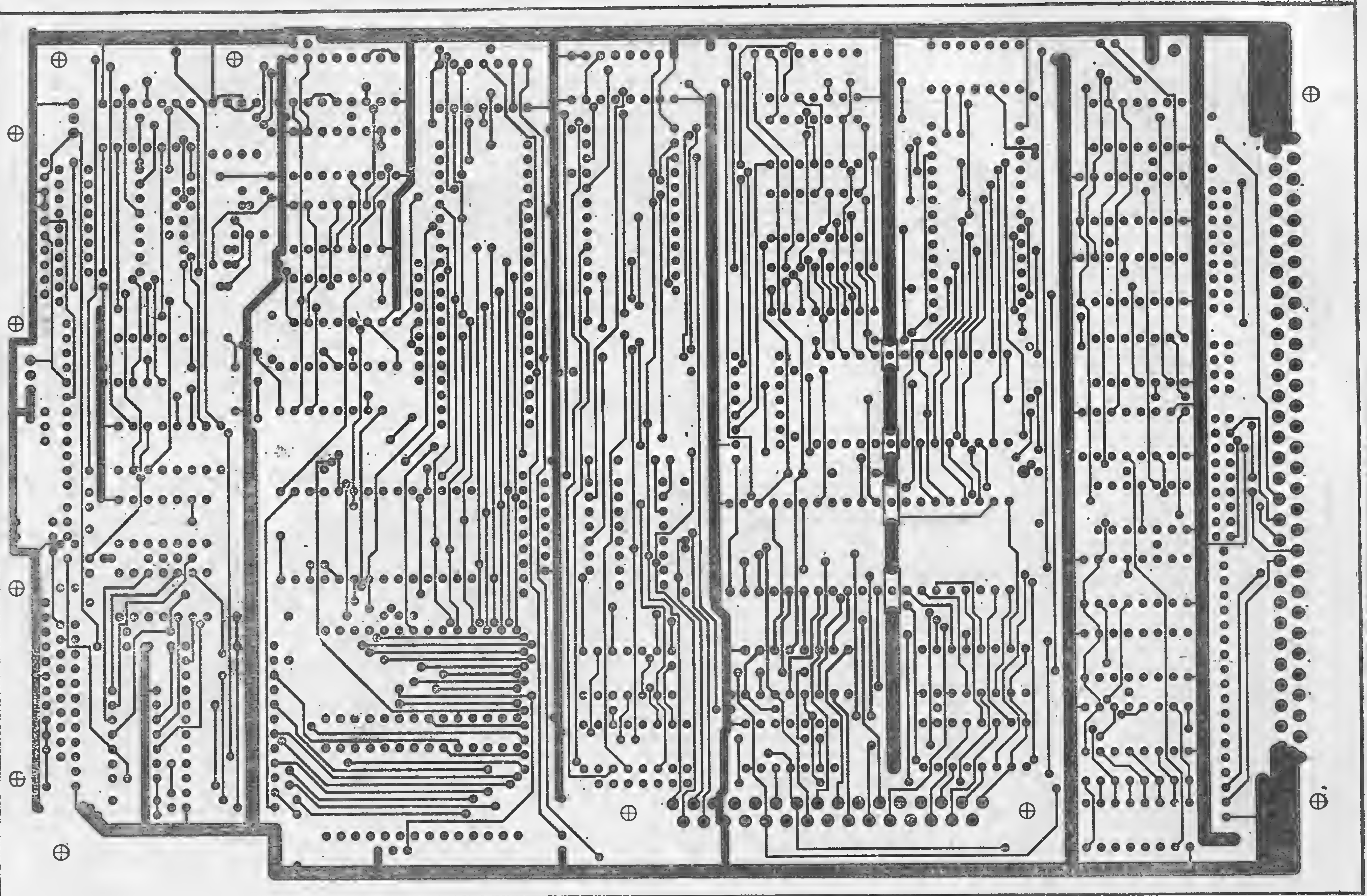


Рис. 12, *а*. Топология, лицевая сторона

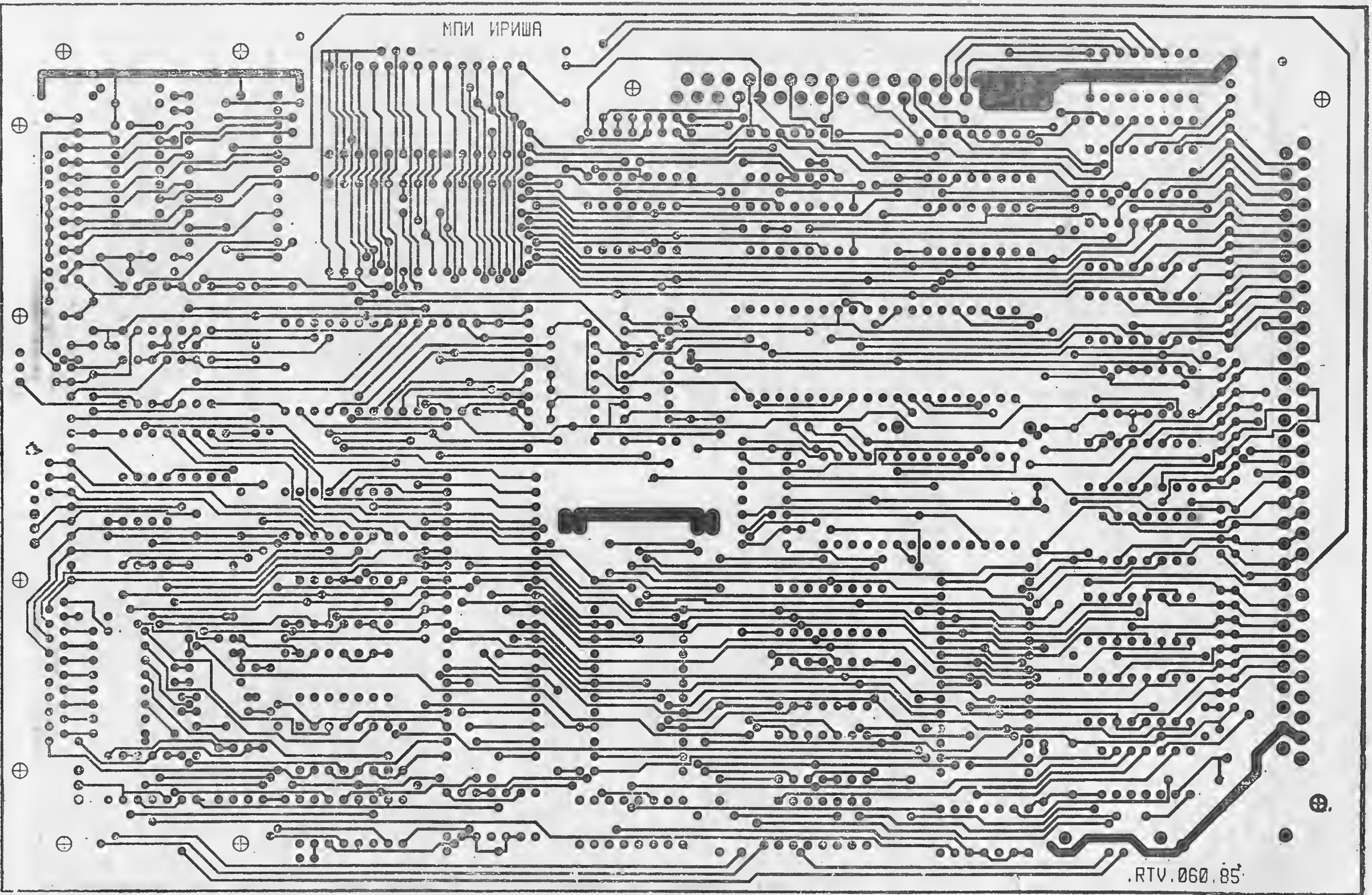


Рис. 12, б. Топология, обратная сторона

ных кодов. Последнее относится только к индикации состояния шины данных и адреса. Кроме устройств индикации, пульт имеет встроенное ПЗУ, содержащее набор наладочных самозацикливающихся программ, и схему формирования синхроимпульсов для наблюдения с помощью осциллографа процесса выполнения отдельных команд. Пульт подключается к технологическому разъему и кроме описанных функций позволяет перевести процессор в режим пошагового выполнения команд.

Модуль индикации магистрали выполняет сходные с наладочным пультом функции по контролю состояния сигналов, но на шинах системной магистрали. Он также позволяет переводить процессор в пошаговый режим обмена данными по системной магистрали.

Тестовые ПЗУ монтируются вместо рабочих в розетки, установленные на плате модуля. Они содержат специальный монитор и ряд тестовых программ, контролирующих работоспособность отдельных узлов модуля. Краткое описание перечисленных устройств будет приведено в последующих номерах журнала.

Статья поступила 27 января 1986 г.

УДК 681.32

Д. А. Лукьянов

«ЭЛЕКТРОНИКА 256К» — ЭМУЛЯТОР ДИСКА ДЛЯ КОМПЛЕКСОВ НА ОСНОВЕ МИКРОЭВМ «ЭЛЕКТРОНИКА 60» И ДВК

В практике научно-технических исследований широко распространены измерительно-вычислительные комплексы, построенные на основе микроЭВМ «Электроника 60» и ДВК. Имея в своем составе накопители на гибких магнитных дисках (НГМД), они позволяют использовать версии простой и удобной операционной системы РАФОС и работать с интерпретирующими системами Фокал и Бейсик. Однако разработка более эффективного и быстродействующего программного обеспечения (ПО) с помощью современных оптимизирующих компиляторов чаще всего затруднена из-за недостаточно большой емкости НГМД, невысокой скорости обмена информацией с ними и недостаточной надежности ввода-вывода.

Для продуктивной и комфортабельной работы с системами Паген/Фортран4, Паскаль2 и Модула2 необходимо около 600, 800 и 1000 информационных блоков на накопителе с ОС и компилятором и 200...1000 свободных блоков для временных файлов. В то же время общий объем НГМД «Электроника 7012» составляет всего около 1000 блоков. Возможность работы с перечисленными компиляторами обеспечит жесткий магнитный диск.

Еще большее быстродействие трансляции можно получить, применив описываемый ниже эмулятор диска, который хранит информацию в дополнительном ОЗУ большой емкости, а в ОС представляется как одно или несколько устройств с файловой структурой прямого доступа. В качестве основы дополнительной подсистемы памяти можно использовать серийные малогабаритные блоки ОЗУ «Электроника 256К», содержащие 256К 16-разрядных ячеек оперативной памяти с непосредственной адресацией и имеющие три независимых канала ввода-вывода информации.

Такому эмулятору не требуется времени на позиционирование магнитных головок, а встроенная система обнаружения и коррекции ошибок информации обеспечивает надежную работу ОС. Работа программиста с эмулятором чрезвычайно проста и состоит в копировании в начале рабочего дня необходимой части ОС и трансляторов с нескольких гибких дисков на эмулятор и загрузки с него ОС. После этой операции НГМД служит только для сохранения подготовленных программ. Вместо НГМД для долговременного хранения ОС и программ может быть использован и другой накопитель, например, на кассетной магнитной ленте.

Аппаратура эмулятора

МикроЭВМ «Электроника 60» с процессором М1 или М2 позволяет непосредственно адресовать только 64К байт программной памяти, поэтому дополнительное ОЗУ строится на внешней (по отношению к процессору) шине обмена информации подсистемы памяти, по набору сигналов аналогичной «Общей шине» СМ ЭВМ (рис. 1). Блоки «Электроника 256К» подключаются к ней непосредственно через имеющиеся интерфейсные платы БМБ-2М, в которых на шину адреса дополнительно выведен разряд адреса А18. Для выбора одного из параллельно включенных блоков ОЗУ они имеют дополнительные линии ЛИП, низкий уровень напряжения на которых блокирует соответствующий блок.

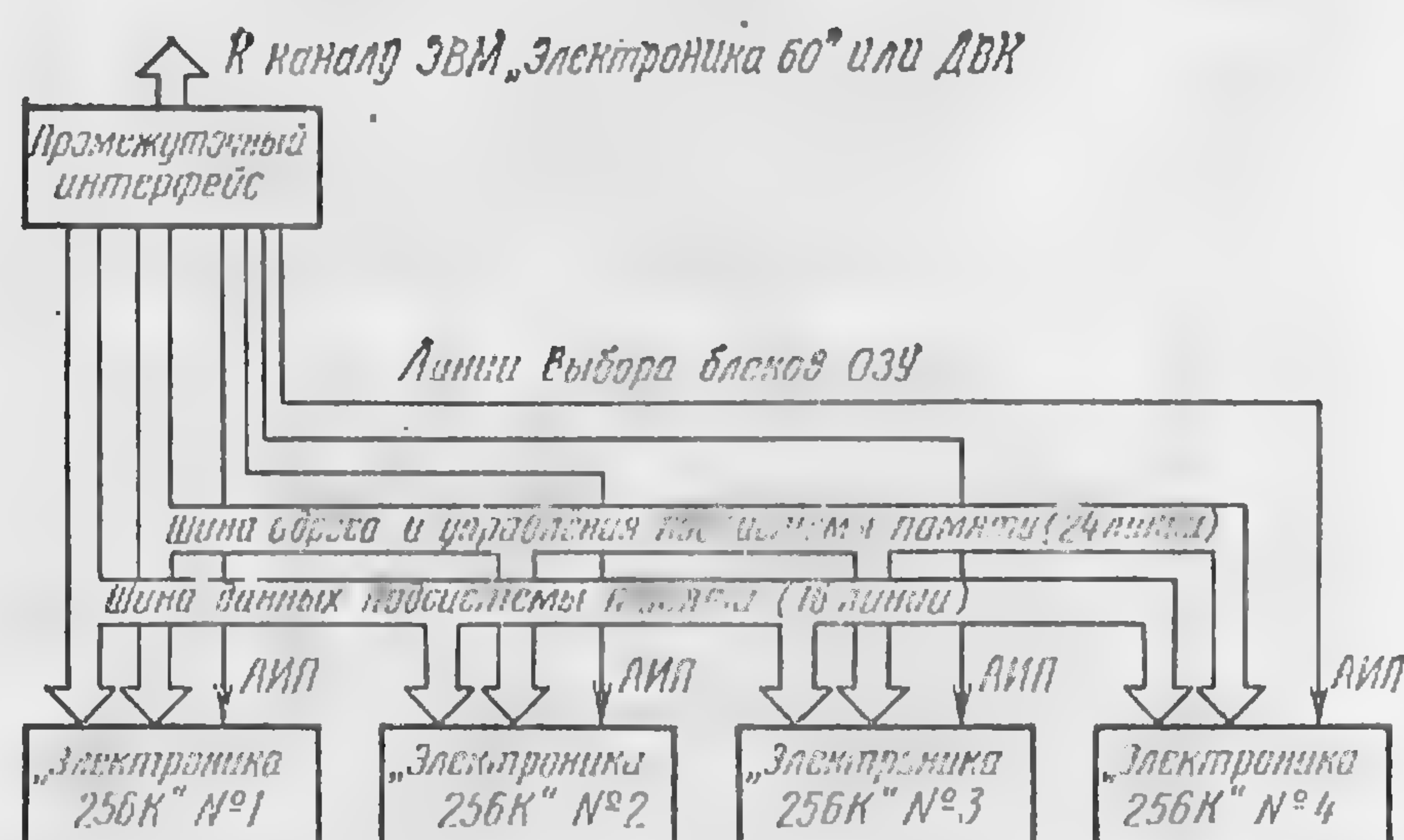


Рис. 1. Структура подсистемы памяти для эмуляции диска объемом 2 Мбайт

Согласование шин подсистемы массовой памяти и канала микроЭВМ происходит с помощью специального промежуточного интерфейса. Так как экономически целесообразно подключение к микроЭВМ такого класса внешнего ОЗУ емкостью не более 2 Мбайт; промежуточный интерфейс обеспечивает адресацию до четырех блоков «Электроника 256К».

Промежуточный интерфейс (рис. 2) при минимальном числе элементов обеспечивает быстрый доступ к

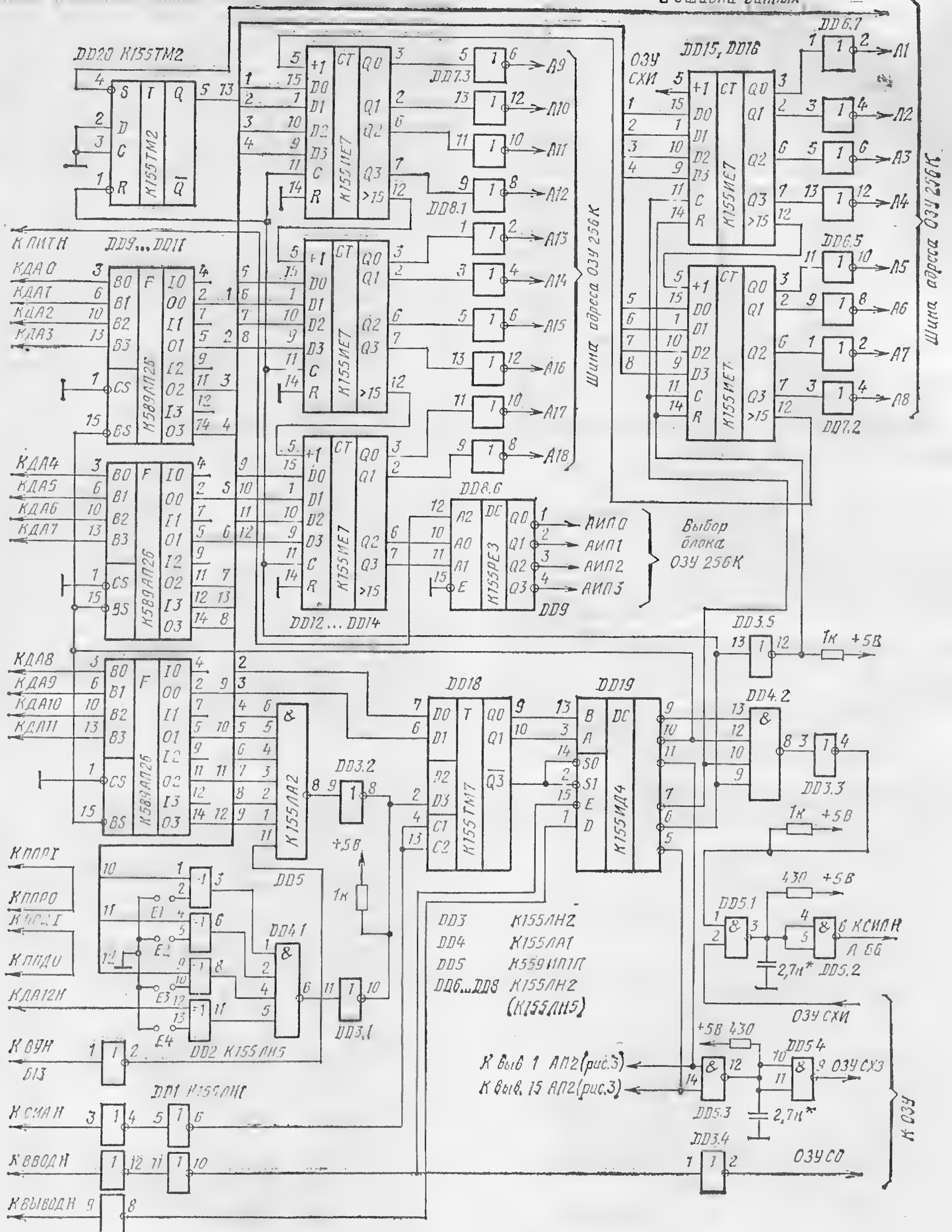


Рис. 2. Принципиальная схема сопряжения канала МПИ и шины адреса подсистемы памяти

Рис. 3. Сопряжение канала «Электроника 60» и шины данных подсистемы памяти

Рис. 4. Сигналы шин и синхронизации промежуточного интерфейса (⁎ — сигналы канала МПИ «Электроника 60»)

любой ячейке или области внешней памяти и требует примитивной программной поддержки операций ввода-вывода, которая возложена на драйвер эмулятора. Интерфейс состоит из канальных приемопередатчиков Q-шины DD9...DD11, логики выбора адреса регистров интерфейса DD1...DD7, регистра младшей части адреса ALOW DD15, DD16, регистра старшей части адреса AHIGH DD12...DD14 и схемы синхронизации.

Регистры интерфейса занимают три последовательных слова адресного пространства страницы ввода-вывода микроЭВМ, причем начало этой области задается

```
.ENABL  LC,GBL

;*************************************************
;* ДРАЙВЕР ЭМУЛЯТОРА ДИСКА В ОС "РАФОС" V4,V5     *
;*************************************************
;* В КАЧЕСТВЕ ВНЕШНЕЙ ПАМЯТИ ИСПОЛЬЗОВАНО ОЗУ     *
;* "ЭЛЕКТРОНИКА-256К" ОБ'ЕМОМ 512 КБАЙТ           *
;*************************************************

.MCALL  .DRDEF,.DREND,.DRFIN,.DRBEG
.DRDEF  EX,0,FILST*,1024.,170770,0

ALOW    =170770             ; АДРЕС РЕГИСТРА ALOW
AHIGH   =170772             ; АДРЕС РЕГИСТРА AHIGH
DATA    =170774             ; АДРЕС РЕГИСТРА ДАННЫХ
                            ; ПОДСИСТЕМЫ ВНЕШНЕЙ
                            ; ПАМЯТИ
EXMASC  =6000               ; МАСКА "НОМЕРА" EX
STATUS  =AHIGH              ; АДРЕС РЕГИСТРА ОШИБКИ

.SBTTL  ВХОД В ДРАЙВЕР В НОРМАЛЬНОМ РЕЖИМЕ

        .DRBEG  EX
        MOV     EXCQE,  R5
        MOV     (R5)+,  R2          ;ВЗЯТЬ НОМЕР СЕКТОРА
        MOV     (R5)+,  R4          ;ПОЛУЧИТЬ "НОМЕР" EX
        TSTB    R4                  ;
        BNE     DONE                ;SPFUN - ИГНОРИРУЮТСЯ
        ASL     R4                  ;ПЕРЕМЕСТИТЬ НОМЕР EX
        ASL     R4                  ;В 10-11 РАЗРЯДЫ

        BIC     #^C<EXMASC>,R4      ;
        ADD     R2,     R4          ;ПОЛУЧИТЬ ФИЗИЧЕСКИЙ
                                    ;АДРЕС НА ШИНЕ ПОД-
                                    ;СИСТЕМЫ ПАМЯТИ
PASSIT: MOV     (R5)+,  R3          ;ПОЛУЧИТЬ АДРЕС БУФЕРА
        MOV     #DATA,  R2          ;ПОЛЬЗОВАТЕЛЯ
        MOV     R4,     @#AHIGH
        MOV     (R5),   R0          ;И СЧЕТЧИК СЛОВ
        BEQ     DONE
        BPL     READIT

WRITIT:                             ;ПЕРЕПИСАТЬ БЛОК
        NEG     R0                  ;ИЗ БУФЕРА ПОЛЬЗОВАТЕЛЯ
        MOVB    R0,     R1          ;ПЕРЕПИСЫВАТЬ ТОЛЬКО
        BIC     #^C<377>,R1         ;ПОЛНЫЕ БЛОКИ, НЕПОЛНЫЕ-
SYC1:                               ;ЗАПОЛНЯТЬ НУЛЯМИ!
10$:    MOV     (R3)+,  @R2
        DEC     R0
        BNE     10$

CLEAR:  CLR     R0                  ;ОЧИСТКА КОНЦА БЛОКА
20$:    TSTB    R1                  ;ЕСЛИ ОН НЕ ЗАПОЛНЕН.
        BEQ     DONE
        MOV     R0,     @R2
        INC     R1
        BR      20$

READIT:                             ;ПРОЧИТАТЬ БЛОК ИЗ
SYC2:   MOV     @R2,    (R3)+       ;ОЗУ В БУФЕР
        DEC     R0                  ;ПОЛЬЗОВАТЕЛЯ
        BNE     SYC2
        TSTB    @#STATUS
        BPL     DONE
ERDONE: MOV     EXCQE,  R5          ;ПРИ ЧТЕНИИ-АНАЛИЗ
        BIS     #HDERR$,@-(R5)      ;НЕКОРРЕКТИРУЕМЫХ
        BR      DONE1               ;ОШИБОК ОЗУ
EXINT:
DONE:                               ;ВОЗВРАТ УПРАВЛЕНИЯ
        MOV     EXCQE,  R5          ;В МОН РАФОС
        CLR     6(R5)
DONE1:  .DRFIN  EX

.SBTTL  ДРАЙВЕР ЭМУЛЯТОРА ДИСКА НА ЭТАПЕ ЗАГРУЗКИ

        ;******* NB:  ЗАГРУЖАЕТСЯ ТОЛЬКО EX0: !
.MCALL  .DRBOT
        .DRBOT  EX,BOOT1,READ
.=EXBOOT+40
BOOT1:  JMP     @#BOOT-EXBOOT
.=EXBOOT+210

READ:   MOV     R0,     @#AHIGH     ;ЗАГРУЗОЧНЫЙ ДРАЙВЕР
SYCR:   MOV     @#DATA, (R2)+       ;ЧИТАЕТ ОБЛАСТЬ СИС-
        DEC     R1                  ;ТЕМНОГО ЗАГРУЗЧИКА
        BNE     SYCR                ;(БЛОКИ 2...5) В ОЗУ
                                    ;И ПЕРЕДАЕТ ЕМУ
TEST:   TSTB    @#STATUS            ;УПРАВЛЕНИЕ.
        BMI     B.ERR
        RTS     PC

B.ERR:  JMP     BIOERR

.=EXBOOT+574
BOOT:   MOV     #10000, SP          ;ИНИЦИАЛИЗАЦИЯ !
        ADD     #3,     R0          ;НАЧАЛО ЗАГРУЗЧИКА
        MOV     #<4*400>,R1         ;ЕГО ДЛИНА (4 БЛОКА)
        MOV     #1000,  R2          ;АДРЕС ЗАГРУЗКИ
        CALL    READ
        MOV     #READ-EXBOOT,@#B$READ
        MOV     #B$DNAM,@#B$DEVN
        CLR     @#B$DEVU
        JMP     @#B$BOOT
        .DREND  EX
.END
```

Рис. 5. Текст программы-драйвера эмулятора диска

перемычками E1...E4. Когда перемычки отсутствуют, регистры ALOW, AHIGH и DATA имеют адреса 170770 170772 и 170774 соответственно.

В ОС РАФОС диски обмениваются с ЭВМ блоками информации, «длина» которых кратна 256 машинным словам, поэтому сегмент внешней памяти, адресуемый кодом, записанным в ALOW, имеет именно такой размер. Отсюда следует, что код старшей части адреса, который записывается в AHIGH, соответствует номеру физического блока «диска» РАФОС. Оба регистра адреса представляют собой части единого счетчика с предварительной установкой начала отсчета, и при каждом обращении к регистру данных промежуточного интерфейса его содержимое увеличивается на единицу, моделируя последовательный доступ к записи на магнитном диске.

Чтобы еще более упростить программу-драйвер эмулятора диска, при записи номера блока в AHIGH одновременно очищается ALOW, подготавливая обращение к началу выбранного блока данных. Если нужна информация из произвольной ячейки подсистемы массовой памяти, то в регистр ALOW следует записать код младшей части необходимого адреса.

Выходы счетчика подключены к шине адреса A1... ...A18 памяти через мощные инверторы DD6...DD8. Два старших разряда счетчика не передаются на шину адреса непосредственно, а поступают на вход ППЗУ DD9, играющего роль дешифратора номера блока «Электроника 256К». Его содержимое таково, что при поступлении сигнала аварии питания КПИТ.Н запрещается обращение ко всем блокам ОЗУ. Логика синхронизации особенностей не имеет и обеспечивает асинхронный обмен данными с квитированием по магистралям подсистемы памяти и микроЭВМ. Внешняя диаграмма обмена показана на рис. 3. Окончание сигнала СХИ ОЗУ инкрементирует содержимое регистров адреса, сразу подготавливая обращение к следующей ячейке памяти. Для уменьшения уровня радиопомех, создаваемых шинами подсистемы памяти, шина данных отключается от магистрали микроЭВМ с помощью буферных приемопередатчиков (рис. 4) в паузах между обращениями.

Включение эмулятора в РАФОС

Для включения эмулятора диска в ОС необходима программа-драйвер (рис. 5), работающая без прерываний программы ЭВМ и обеспечивающая передачу одного блока информации примерно за 2 мс. Каждый блок ОЗУ (см. рис. 1) представлен в ОС как устройство с именем EX⟨N⟩ ($N=0$, 1, 2, 3) объемом 1010 информационных блоков, имеющее свой каталог файлов. Все программные запросы к EX аналогичны запросам к другим устройствам прямого доступа ОС, но удовлетворяются немедленно. Драйвер EX состоит из двух секций: рабочего драйвера и драйвера этапа загрузки. Последний позволяет запускать ОС с эмулятора диска, если в него записаны монитор и сервисные программы.

Для работы с EX служат следующие команды:

INIT EX⟨N⟩ — инициализация каталога эмулятора после включения питания;

COPY/BOOT — копирование системного загрузчика в соответствующую область внешнего ОЗУ;

EX0:RAFSJ EX0:

BOOT EX0: — загрузка операционной системы.

Текст драйвера необходимо оттранслировать и включить в операционную систему командами:

```
MACRO/OBJ:EX EX+SYSMAC.SML/LIBRARY
LINK/EXE:SY:EX.SYS EX
INSTALL EX:
INIT/NOQUERY EX0: (EX1:, EX2:, ...).
```

Дальнейшее развитие системы

Блоки ОЗУ «Электроника 256К», подключенные описанным способом к микроЭВМ, открывают ряд дополнительных возможностей. Во-первых, с одной подсистемой массовой памяти можно использовать до трех микроЭВМ. Во-вторых, появляется возможность копирования ОС в эмулятор из другой (центральной) ЭВМ, что удобно в системах, ориентированных на обучение программированию, где на длительное время требуется сохранять лишь малую часть введенных программ. Для передачи информации в таком случае можно воспользоваться либо уже имеющимся в комплексах ДВК последовательным интерфейсом, либо применить специальный связной процессор, например, на базе МПК БИС серии КР580. В-третьих, вместо одного из магистральных буферов в соответствующий разъем ОЗУ может быть установлена плата вывода информации на экран телевизора в графическом виде.

Статья поступила 20 апреля 1985 г.

ПОПРАВКИ ПО ЖУРНАЛУ «МП» № 1, 1986.

1. В статье Романова В. Ю. и др. (с. 70) вместо приведенного рис. 11 — Карта прошивки PC. GD.SA1 ПЗУ D3 селектора адреса для внутренней памяти объемом 16К байт (*а*) и 64К байт (*б*) — должен быть рисунок, приведенный ниже.

а)

ADRESS (HEX) X=	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	- DATA (HEX) -															
0X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2X	1	9	4	C	2	A	6	E	1	9	4	C	2	A	6	E
3X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6X	1	9	4	C	2	A	6	E	1	9	4	C	2	A	6	E
7X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
AX	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
BX	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CX	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DX	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
EX	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FX	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

б)

ADRESS (HEX) X=	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	- DATA (HEX) -															
0X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2X	1	9	5	D	3	B	7	F	1	9	5	D	3	B	7	F
3X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6X	1	9	5	D	3	B	7	F	1	9	5	D	3	B	7	F
7X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
AX	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
BX	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CX	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DX	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
EX	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
FX	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

2. На страницах 12 и 31 рисунки 5 переставлены местами.

3 Заказ № 4 7

УДК 681.3.05

А. А. Томчук, И. В. Музалевский, Т. В. Вдовыко,
В. Н. Невзоров

МОДУЛЬ СЧЕТЧИКА-ТАЙМЕРА В СТАНДАРТЕ МИКРОЭВМ «ЭЛЕКТРОНИКА 60»

Типичным для многих систем обработки данных специализированных и технологических контроллеров на базе универсальных микроЭВМ является временной анализ информации, широко используемый в физическом эксперименте. А для управления внешними объектами в таких системах возникает необходимость в формировании последовательностей импульсов с заданными временными соотношениями, модифицируемыми в широких пределах в процессе работы (например, для управления шаговыми двигателями).

В настоящее время из-за отсутствия в составе существующих МП систем и микроЭВМ модулей, обеспечивающих частотно-временное преобразование кодовой информации, большинство пользователей вынуждено проектировать свои специальные устройства сопряжения [1—5]. Эти устройства, кроме дублирования основных функций, отличаются узким диапазоном возможных применений и излишними аппаратурными затратами.

Далее рассматриваются структура программно-управляемого модуля счетчика-таймера, выполненного в стандарте микроЭВМ «Электроника 60» на основе БИС КР580ВИ53, и основные процедуры взаимодействия с ним. Таймер КР580ВИ53 содержит три 16-разрядных вычитающих счетчика с независимым программным управлением их состояниями и режимами работы [6, 7].

Последовательность обращений к счетчикам и соответствующим регистрам управления не имеет значения, но при операциях с полным 16-разрядным содержимым счетчиков обязательно должны быть произведены два обращения: в первом осуществляется доступ к содержимому младшего байта выбранного счетчика; во втором—к старшему. Именно это обстоятельство, а также некоторые особенности в последовательности управляющих сигналов канала микроЭВМ «Электроника 60» (наличие сигнала К БАЙТ Н в цикле «ввод — модификация — вывод» при выполнении ряда команд) необходимо учитывать при написании программ управления модулем.

Как показано на электрической схеме (рис. 1), модуль содержит четыре БИС таймера (DD7...DD10), программный доступ к которым организован через 8-разрядную внутримодульную шину данных, интерфейсную часть схемы (DD1...DD4; DD23...DD25), а также внутренний генератор опорной частоты (DD27; GN1) с F=2 МГц.

Для расширения функциональных возможностей модуля в его состав введены регистры масок (DD12, DD14; DD13, DD15), позволяющие управлять схемами выходных усилителей — разрешение выдачи (РВ), а также программно управлять входами разрешение счета (PC) таймеров (ТО...Т3).

Если требуется программный контроль состояния входов PC счетчиков, необходимо установить на плате перемычку между соответствующим входом таймера и выходом регистра маски. В противном случае предполагается, что управление входами PC осуществляется внешними схемами.

Модуль счетчика-таймера обеспечивает прием входных сигналов, согласованных с ТТЛ-

Распределение адресов модуля

Смещение относительно базового адреса	Тип обращения	
	Словные к счетчику N	Байтовые к регистру маски
0 (0) 2 (2) 4 (4)	0 1 } T0 2	— —
6 (6)	РУС Т0	PC0
10 (8) 12 (A) 14 (C)	3 4 } T1 5	— —
16 (E)	РУС Т1	PC1
20 (10) 22 (12) 24 (14)	6 7 } T2 8	— —
26 (16)	РУС Т2	РВ0
30 (18) 32 (1A) 34 (1C)	0 A } T3 B	— —
36 (1E)	РУС Т3	РВ1

Примечания: – – — неиспользуемые форматы обращений; в скобках — 16-ричный код; РУС — регистр управляющего слова.

Резисторы: (R1...R8) – МЛТ – 0,125 – 1 кОм ± 20%
(R9...R12) – МЛТ – 0,125 – 300 Ом ± 20%

Конденсаторы: C1 КМ6 – 1,0 мкФ – ± 10%
C2 – К53-3 – 220 мкФ × 6В ± 10%
C3 – КМ-5а – 390 пФ ± 10%
C4 – КМ-5а – 2700 пФ ± 10%

Микросхемы: (DD1...DD4) – К589АП26;
DD5, DD6 – К589АП16; (DD7...DD10) – КР580ВИ53; (DD11...DD13) – К589ИР12;
DD14, DD15 – К155ТМ8; (DA16...DA18) – К131НР1-2,2 кОм; (DD19...DD21) – К155ЛА11;
DD22 – К556РТ4; DD23 – К556РТ5; DD24, DD27 – К155ЛН1; DD25 – К555ЛН1;
DD26 – К155ЛН2; DD28 – К556РТ4; DD29 – К155ЛЕ1.
GN1 – резонатор кварцевый – 2000 кГц.

Рис. 1. Электрическая схема модуля счетчика-таймера

уровнями, с частотой следования до 2 МГц. Выходные усилители (DD19...DD21) с выходом типа открытый коллектор непосредственно управляют высоковольтными схемами (до 15 В) без дополнительного преобразования уровня сигнала.

В модуле, содержащем 12 16-разрядных вычитающих счетчиков, адреса распределены, как указано в приведенной таблице.

Особенность адресного распределения — использование адресов устройства для обращения к регистрам таймеров и к регистрам масок в словарном и байтовом форматах соответственно. Поскольку и те и другие регистры программно доступны только для записи, то для дешифрации обращения к той или другой аппаратной единице целесообразно использовать соответственно словные или байтовые форматы команд микроЭВМ. Это позволяет сэкономить адресное пространство, занимаемое модулем. Так, при выделении отдельных адресов обращений к регистрам масок модуль занимал бы в общей сложности 18 ячеек (вместо 16), что с системотехнической точки зрения представляется не вполне оправданным. Кроме того, учитывалась схемотехническая сложность устройства и сложность трассировки печатной платы. Этим объясняется доступ через общую внутримодульную шину данных как к счетчикам, так и к регистрам масок (рис. 2). Запись «Лог. 1» в любой из разрядов регистров масок разрешает работу соответствующих счетчиков и открывает соответствующий выходной усилитель.

Номер регистр.	Смещение РС	Смещение РВ	7	6	5	4	3	2	1	0	Разряды данных
0	6(6)	26(16)	7	6	5	4	3	2	1	0	Номера счетчиков
1	16(E)	36(1E)					B	A	9	8	

Рис. 2. Формат управляющего байта регистров масок

Базовый адрес модуля записывается в ППЗУ интерфейсного слоя и может модифицироваться с помощью перемычек, устанавливаемых на плате.

Для того, чтобы организовать на основе модуля счетчика-таймера управление конкретным объектом, необходимо в первую очередь определить конфигурацию внешних связей, режимы работы каждого счетчика, а также их связи друг с другом с помощью перемычек на выходном разъеме. Так, счетчики, входящие в состав модуля, могут каскадироваться по счетным входам СК и по входам РС [6]. Это позволяет существенно расширить разрядность обрабатываемых данных и ввести, таким образом, дополнительные аппаратурные реализуемые функции управления одного счетчика другим.

Затем, в соответствии с выбранной конфигурацией, загружаются регистры масок, а также регистры управления используемых таймеров. Инициализация модуля завершается записью начального состояния в счетчики, после чего счетчик-таймер начинает работать в режиме контроллера, формирующего заданные последовательности импульсов или осуществляющего счет импульсов, поступающих от объекта. При этом режим «защелка» используется для считывания содержимого счетчиков, обеспечивая считывание истинных отсчетов без остановки процесса счета.

Контроль за состоянием модуля осуществляется либо программным опросом, либо через прерывание по внешнему событию (см. рис. 1). При этом следует учитывать особенности схемы КР580ВИ53:

1. Внутреннее содержимое счетчиков продолжает модифицироваться и после формирования сигнала достижения границы счета на его выходе. Это делает невозможным фиксацию программными средствами факта наступления события (например, в режиме таймера). Исключить указанный недостаток можно, организовав чтение состояния выходов счетчиков. Состояния выходов восьми младших счетчиков (0...7) считываются в цикле Ввод в старшем байте слова при обращении по любому из адресов модуля.

2. Требуется использование дополнительных схемотехнических и программных приемов в конкретных применениях модуля счетчика-таймера. Например, всякий раз после достижения значения 0 счетчики восстанавливают свое первоначальное содержимое и продолжают его декрементировать. Поэтому, если необходимо произвести N отсчетов, где $N=2^{16}$, оказывается невозможным простое каскадирование необходимого числа счетчиков и задание их исходных значений непосредственно в двоичном виде, так как фактический коэффициент пересчета младших счетчиков не будет равен 2^{16}.

3. Необходимо подавать хотя бы один отрицательный перепад напряжения на тактовые входы счетчиков для начала их работы в соответствии с запрограммированным режимом. Это приводит к тому, что приходится учитывать наличие лишнего импульса на выходе модуля (если это необходимо). Например, если один счетчик используется в режиме генерации заданного числа управляющих импульсов, подсчет которых осуществляется вторым счетчиком, то внешний выход первого счетчика блокируется регистром маски на время прохождения первого импульса. В режиме счета,

наоборот, на входе счетчика генерируется дополнительный импульс инициализации, чтобы избежать считывания ложных данных в случае, если на вход не поступит ни одного информационного сигнала.

Общее потребление тока от источника напряжения 5 В около 2,5 А. Конструктивно модуль выполнен в формате 1/2 платы микроЭВМ «Электроника 60».

Рассмотренный модуль счетчика-таймера используется в системе управления точными линейными перемещениями с помощью шаговых двигателей (ШД) типа ШДР-711 и ДШИ-200. Он позволяет организовать независимое управление четырьмя механическими приводами на основе ШД, исключить просчет шагов ШД и значительно упростить построение системы точного контроля положения привода за счет введения канала обратной связи. Структурная схема такого устройства (рис. 3) состоит из формирователя временных последовательностей (ФВП) и каналов управления (КУ).

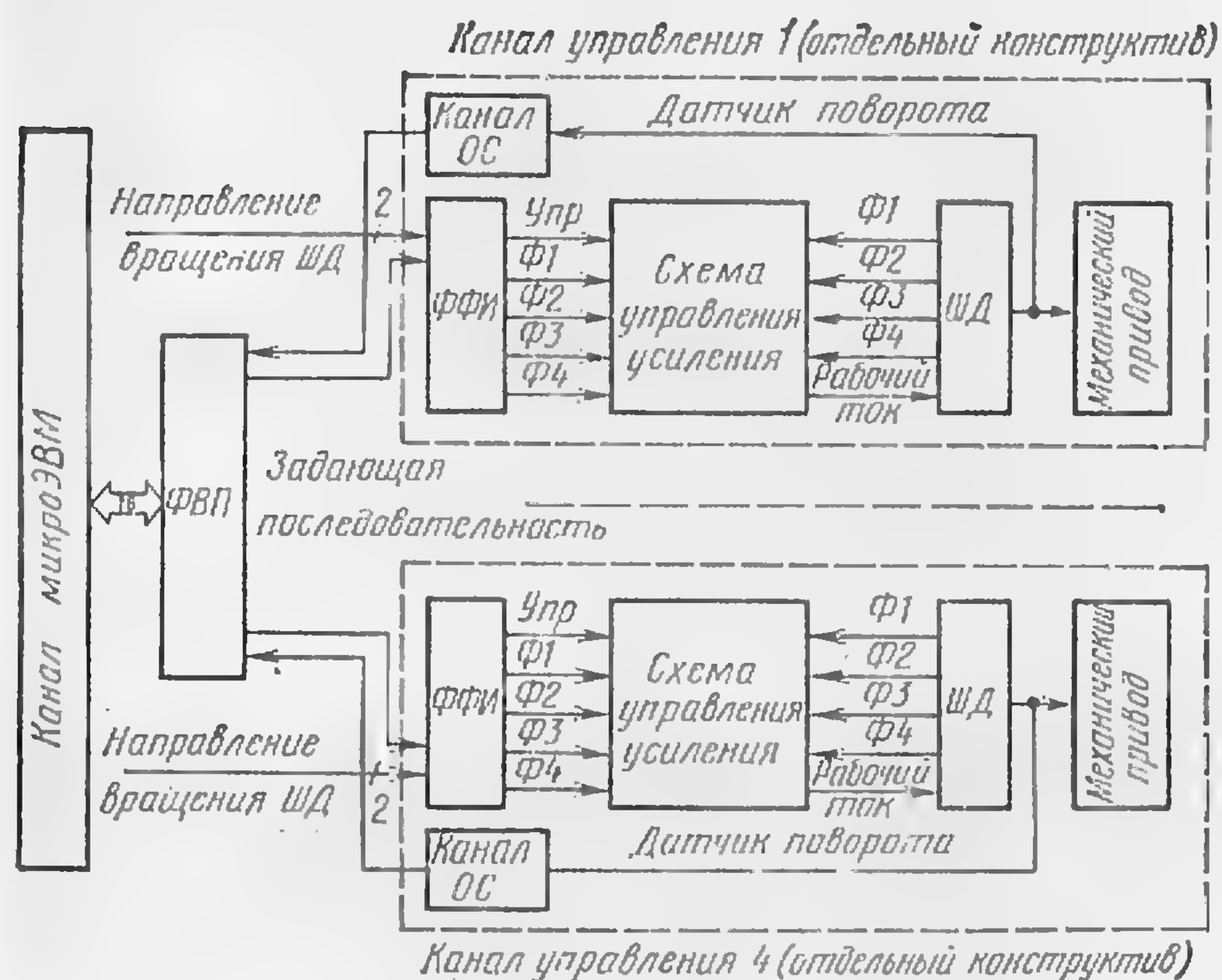


Рис. 3. Структурная схема управления шаговыми двигателями в системе точных линейных перемещений

В функции ФВП входят: прием от микроЭВМ кодов значений перемещения и скорости вращения ШД для каждого КУ; выдача на формирователь фазных импульсов (ФФИ) задающей последовательности с частотой и числом импульсов, соответствующих скорости вращения ШД и перемещению; прием и обработка сигналов канала ОС; прием и выполнение текущих команд от микроЭВМ на приостановку и возобновление движения.

Функции ФВП обеспечиваются модулем счетчика-таймера при задании необходимой конфигурации счетчиков с соответствующими коммутациями на выходном разъеме модуля без изменения его внутренних связей (рис. 4).

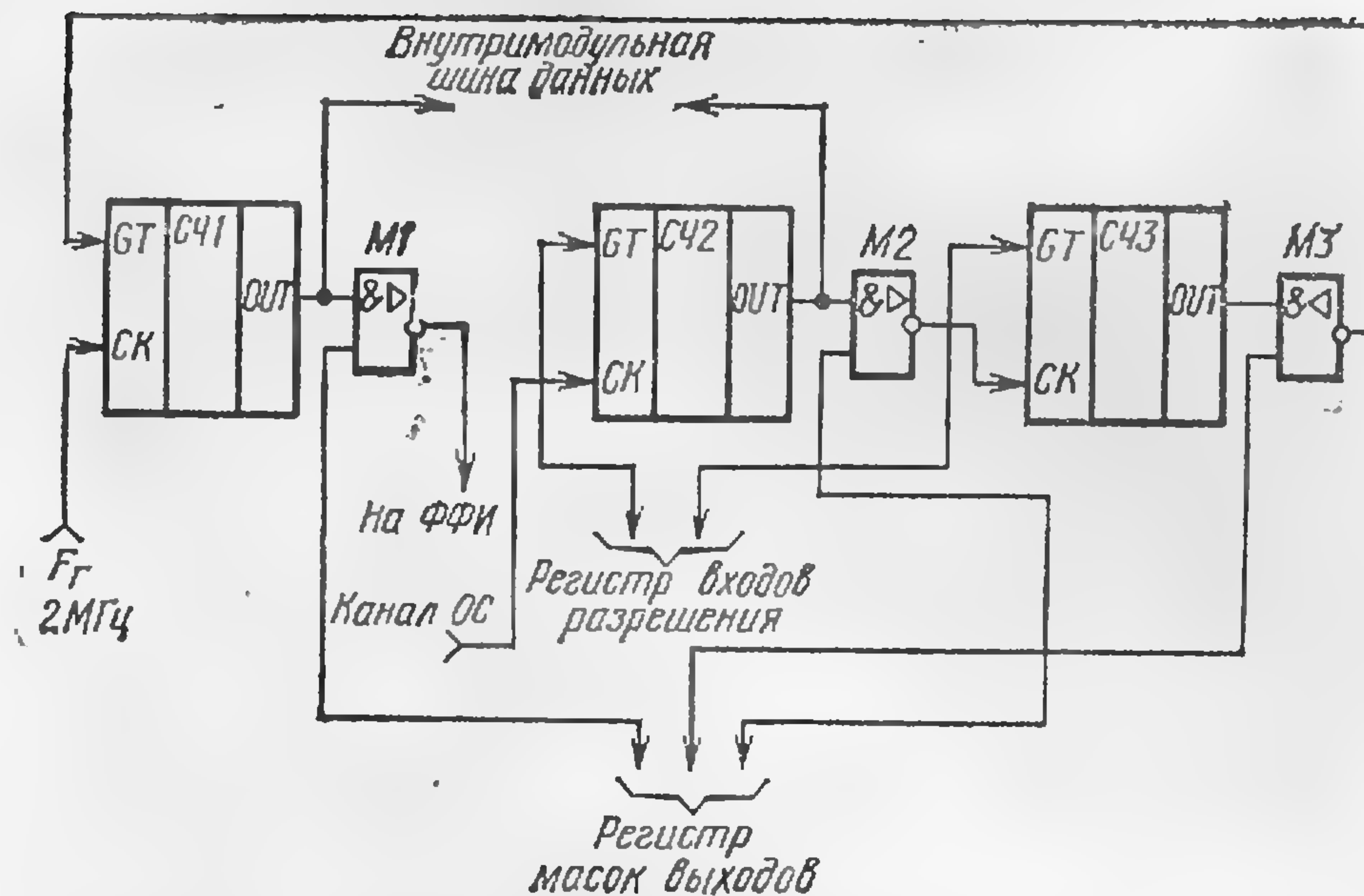


Рис. 4 Конфигурация счетчика-таймера, реализующая функции ФВП: GT — вход РС; СК — тактовый вход; OUT — выход

В случае, если счетчики СЧ1...СЧ3 принадлежат таймеру Т0 и имеющие адреса в соответствии с таблицей, то СЧ1 формирует задающую последовательность требуемой частоты и работает в режиме 3 (делитель частоты). Счетчик 2 также работает в режиме 3 и использован для повышения коэффициента пересчета. Счетчик 3 работает в режиме 1 (0-таймер) и управляет работой генератора СЧ1, останавливая его после выдачи заданного числа импульсов. Программа инициализации приведена на рис. 5.

```
                                          / PAGE 0001. / TOTAL 0001.

                       АССЕМБЛЕР MACRO-11 / РАФОС
**************************************************************************
*   ПРИМЕР ПРОГРАММЫ ИНИЦИАЛИЗАЦИИ СЧЕТЧИКОВ В КОНФИГУРАЦИ ,            *
*   ВЫПОЛНЯЮЩЕЙ ФУНКЦИИ КОНТРОЛЛЕРА ШД .                                 *
*   ПРИ ВЫЗОВЕ ПРОГРАММЫ НЕОБХОДИМО ПОДГОТОВИТЬ СЛЕДУЮЩИЕ               *
*   ВХОДНЫЕ ДАННЫЕ :                                                      *
*   В РЕГИСТРЕ R1 - КОД ВЕЛИЧИНЫ ПЕРЕМЕЩЕНИЯ ,                           *
*   В РЕГИСТРЕ R2 - КОД КОЭФФИЦИЕНТА РЕДУКЦИИ МЕХАНИЧЕСКОГО              *
*                   ПРИВОДА     ,                                         *
*   В РЕГИСТРЕ R3 - КОД ВЕЛИЧИНЫ СКОРОСТИ                                *
**************************************************************************
.PSECT  INIT
INIT:   MOV     #160000,R0          ;В РЕГИСТРЕ R0 НАХОДИТСЯ БАЗО-
                                    ;ВЫЙ АДРЕС УСТРОЙСТВА
                                    ;ПРОЦЕДУРА НАЧАЛЬНОГО СБРОСА
                                    ;СИСТЕМЫ
        MOVB    #0,6(R0)            ;РАБОТА СЧЕТЧИКОВ ЗАПРЕЩЕНА
        MOVB    #0,16(R0)           ;ПО ВХОДАМ " РС "
        MOVB    #0,26(R0)           ;ВЫХОДНЫЕ ДРАЙВЕРЫ ЗАКРЫТЫ
        MOVB    #0,36(R0)
        MOV     #66,6(R0)           ;ЗАПИСИ В " РУС " КОМАНДЫХ
        MOV     #166,6(R0)          ;СЛОВ , ОПРЕДЕЛЯЮЩИХ РЕЖИМ
        MOV     #266,6(R0)          ;РАБОТЫ СЧЕТЧИКОВ И ФОРМАТЫ
                                    ;СЛОВ ДАННЫХ
        MOV     R3,(R0)             ;КОЭФФИЦИЕНТ ДЕЛЕНИЯ СЧ1
        SWAB    R3                  ;( СКОРОСТЬ ВРАЩЕНИЯ РОТОРА
        MOV     R3,(R0)             ;ШД )
        MOV     R2,2(R0)            ;КОЭФФИЦИЕНТ РЕДУКЦИИ МЕХАНИ -
        SWAB    R2                  ;ЧЕСКОГО ПРИВОДА
        MOV     R2,2(R0)
        MOV     R1,4(R0)            ;ВЕЛИЧИНА ПЕРЕМЕЩЕНИЯ
        SWAB    R1
        MOV     R1,4(R0)
        MOVB    #6,6(R0)            ;РАЗРЕШЕНА РАБОТА СЧЕТЧИКОВ
                                    ;СЧ2 И СЧ3 НУЛЕВОГО ТАЙМЕРА
        MOVB    #7,26(R0)           ;ОТКРЫТЫ ВЫХОДНЫЕ ДРАЙВЕРЫ.
                                    ;РАЗРЕШЕНА. РАБОТА СЧ1 НУЛЕВОГО
                                    ;ТАЙМЕРА

        RETURN                      ;НАЧАЛОСЬ ДВИЖЕНИЕ
```

Рис. 5. Программа INIT

Для контроля положения механического привода может быть использовано ОЗУ с автономным батарейным питанием в режиме хранения информации. Его содержимое модифицируется микроЭВМ в соответствии с совершаемыми перемещениями, что позволяет сохранить информацию о реальном положении системы в случае отключения питания. Построенная схема управления ШД позволяет осуществить независимые перемещения четырех приводов в диапазоне $-(2^{32}-1)...+$ $+(2^{32}-1)$ шагов и реализует все необходимые режимы работы ШД.

Рассмотренное устройство можно использовать и для решения других задач автоматизации и управления техническими системами с использованием микроЭВМ. В частности, задачи возникают при построении устройств, предназначенных для радио- и спектрометрии излучений. Например, при геофизических исследованиях скважин импульсными радиоактивными методами идентификация характеристик пород осуществляется по показаниям интенсивностей счета регистрируемых частиц в последовательно следующих друг за другом временных интервалах с изменяемой в процессе работы длительностью.

Для справок обращаться: 115409, г. Москва, Каширское шоссе, дом 31, МИФИ, каф. «Электроника».

ЛИТЕРАТУРА

1. Бородин С. М., Томчук А. А. Средства отладки и сопровождения в системе обработки данных для импульсного нейтронного каротажа скважин. — В кн.: Малые ЭВМ и системы автоматизации научных исследований. — М.: Энергоатомиздат, 1983, с. 79.

2. Бялко А. А., Нистратов А. Н., Смит В. Ю., Чураков А. К. Гамма-спектрометрический комплекс на базе ЭВМ СМ-4 для нейтронно-активационного анализа радионуклидов. — В кн.: Малые ЭВМ и системы автоматизации научных исследований. — М.: Энергоатомиздат, 1983, с. 65.

3. Меликов Н. Ю. Цифровое устройство обработки сигналов лазерного газоанализатора. — В кн.: Электронные приборы и схемы для экспериментальной физики. — М.: Энергоатомиздат, 1983, с. 32.

4. Зеленин В. П., Карпов С. Б., Ярославцева В. В. Программируемый генератор импульсов. — ПТЭ, 1982, № 4, с. 83—87.

5. Соколовский В. Ш., Тюрин В. А., Курнеев В. В. Генератор импульсных последовательностей с программным управлением для ЯМР. — ПТЭ, 1983, № 3, с. 124—126.

6. Алексенко А. Г., Галицын А. А., Иванников А. Д. Проектирование радиоэлектронной аппаратуры на микропроцессорах. — М.: Радио и связь, 1984, с. 65—71.

7. Торгов Ю. И. Программируемый таймер КР580ВИ53 и его применение. — Микропроцессорные средства и системы, 1984, № 1, с. 77—84.

Статья поступила 15 ноября 1985 г.

Окончание. Начало см. на стр. 9

Структурная схема К583ВА4 (см. рис. 8) содержит:
четыре 8-разрядных двунаправленных магистрали L1 [0...7] ... L4 [0...7] с выходом с «открытым» коллектором;
четыре 8-разрядных буферных регистра R61...R64, синхронизируемых уровнем;
четыре 8-разрядных логических устройства LU1... ...LU4;
12-разрядную магистраль управления S1...S13;
одноразрядную магистраль синхронизации STB.

Логические устройства LU1...LU4 асинхронно выполняют логические операции над операндами, поступающими с магистралей L1...L4 (табл. 6). Результат операции логических устройств поступает в регистры данных R61...R64 или непосредственно на магистрали L1...L4.

Микросхема выполняет операции мажоритирования байтовых данных для трех операндов.

Работа буферных регистров микросхемы R61...R64 синхронизирована одним синхросигналом STB.

Перепад уровней сигнала STB стробирует занесение информации во все регистры данных R61...R64 одновременно.

Для микросхемы возможны два режима работы: при значении управляющего сигнала STB=0 в регистрах R61...R64 результат выполнения логических операций в LU1...LU4 не сохраняется; при значении управляющего сигнала STB=1 регистры R61...R64 и магистрали L1...L4 сохраняют результат выполнения логических операций в логических устройствах.

Статья поступила 3 сентября 1985 г.

РЖ АВТ-86

1Б270. **Операционные системы для персональных ЭВМ.** Von CP/M zu Unix — Betriebss. Systemm für personal computer. „Computer persönlich", 1985, № 16, 60—63 (нем.)

Рассматривается развитие операционных систем для персональных ЭВМ. Первой появилась операционная система CP/M, которая предназначалась для ЭВМ с 8-разрядными МП типов Z80, 8080, 8086 и с объемом ОЗУ до 64К байт. Операционная система CP/M состоит из следующих модулей: CCP — интерпретатор команд, BDOS — базовая дисковая операционная система, BIOS — базовая система ввода-вывода и служебные команды.

В настоящее время семейство CP/M состоит из следующих версий: наиболее распространенной CP/M 2.2. или CP/M-80 для ЭВМ с 8-разрядными МП и ОЗУ на 64К байт; CP/M Plus, которая может работать и с большими объемами ОЗУ; CP/M для ЭВМ с 16-разрядными МП типов 8088 и 8086; CP/M-68K для ЭВМ с МП типа 68000, MP/M и MP/M-86, Concurrent DOS и Concurrent DOS-286 для ЭВМ с МП типа Intel 8086 и 80286.

При появлении ЭВМ с 16-разрядными МП фирма Microsoft разработала для них операционную систему MS—DOS. Указывается, что в настоящее время все большее распространение для персональных ЭВМ приобретает операционная система Unix.

Программируемые логические матрицы (ПЛМ) являются удобной формой кремниевой реализации набора произвольных функций алгебры логики, настраиваемых разработчиком. ПЛМ «наводят мосты» между микропроцессорами и специализированными БИС, успешно заменяя большое число схем произвольной логики.

УДК 681.326.3

Н. Н. Щелкунов, А. П. Дианов

ПРОЦЕДУРЫ ПРОГРАММИРОВАНИЯ ЛОГИЧЕСКИХ МАТРИЦ

Программируемая матричная логика (ПМЛ) незаменима при реализации функций И-НЕ, ИЛИ-НЕ, НЕ и т. д., необходимых при сопряжении БИС между собой. Использование приборов ПМЛ позволяет разработчикам микропроцессорных систем самостоятельно занести и отредактировать информацию для выполнения конкретной функции, что значительно облегчает процесс конструирования систем, придает им дополнительную гибкость.

Существуют однократно программируемые биполярные матричные БИС с выжигаемыми связями [1, 2] и многократно программируемые ПМЛ с ультрафиолетовым стиранием на МОП-структурах с плавающими затворами. Микросхемы ПМЛ могут быть выполнены по КМОП-технологии [3—5] и иметь электрически перестраиваемую структуру [6], т. е. любая технология программируемых приборов памяти может быть использована в этих микросхемах. Приборы ПМЛ насчитывают три вида: программируемые вентильные матрицы (ПВМ), программируемые логические матрицы (ПЛМ) и программируемые логические контроллеры (ПЛК) [1]. Среди них наиболее популярны ПЛМ. Широко используются однократно программируемые ПЛМ с нихромовыми плавкими перемычками КР556РТ1 и КР556РТ2 [7], выполненные по ТТЛШ-технологии. Микросхемы имеют одинаковую логическую структуру и отличаются только типом выхода: микросхема КР556РТ2 имеет тристабильный выходной каскад, а КР556РТ1 — выходной буфер с открытым коллектором. Условное графическое обозначение этих ПЛМ представлено на рис. 1, а ниже приведены основные параметры ПЛМ серий К556 и КР556.

	КР556	К556
Напряжение питания, В	5±5%	
Ток потребления, мА	⩽170	⩽180
Входной ток «Лог. 0», мкА	100	
Входной ток «Лог. 1», мкА	25	
Выходной ток «Лог. 0», А	20...70	
Входная емкость, пФ	5	
Выходная емкость, пФ	8	
Емкость нагрузки, пФ	⩽150	
Время распространения от адресного входа, нс	70	80
Время распространения от $\overline{CE}$, нс	30	45
Тип корпуса	2121.28-1	4119.28-1

Микросхемы реализуют восемь функций $F_{j,j}=\overline{0,7}$ от 16 входных переменных $A_{m,m}=\overline{0,15}$. При этом если логические функции представить в дизъюнктивной нормальной форме, то общее число конъюнкций для всех функций не должно превышать 48. Микросхемы имеют инверсный вход разрешения кристалла $\overline{CE}$ и линию разрешения программирования FE.

Эквивалентная логическая схема ПЛМ содержит два уровня программируемой логики (рис. 2). П е р в ы й л о г и ч е с к и й у р о в е н ь включает матрицу из 48 схем И (конъюнкторов), организующую логические произведения (конъюнкции) $P_{i,i}=\overline{0,47}$ от входных переменных $A_{m,m}=\overline{0,15}$ и их инверсий $\bar{A}_{m,m}=\overline{0,15}$:

$$P_i=\prod_{m=0}^{15}(\alpha_{im}A_m+\overline{\alpha_{im}})(\alpha^*_{im}\bar{A}_m+\overline{\alpha^*_{im}})=$$

$$=\prod_{m=0}^{15}(\overline{\alpha^*_{im}}A_m+\overline{\alpha_{im}}\,\overline{A_m}),\quad i=\overline{0,47}.$$

Инвертирование логических переменных $A_{m,m}=\overline{0,15}$ осуществляется входными буферными усилителями. В исходном состоянии все плавкие связи сохранены ($\alpha_{im}=\alpha^*_{im}=1$, $i=\overline{0,47}$, $m=\overline{0,15}$), что обеспечивает нулевое значение всех конъюнкций. С помощью выплавляемых связей α_{im} и α^*_{im}, $i=\overline{0,47}$, $m=\overline{0,15}$, каждый i-конъюнктор может быть соединен либо непосредственно с входной переменной A_m ($\alpha_{im}=1$, $\alpha^*_{im}=0$), либо с ее инверсией $\bar{A}_m$ ($\alpha_{im}=0$, $\alpha^*_{im}=1$). Удаление перемычек ($\alpha_{im}=\alpha^*_{im}=0$) устанавливает независимость конъюнкции P_i от переменной A_m. Не до-

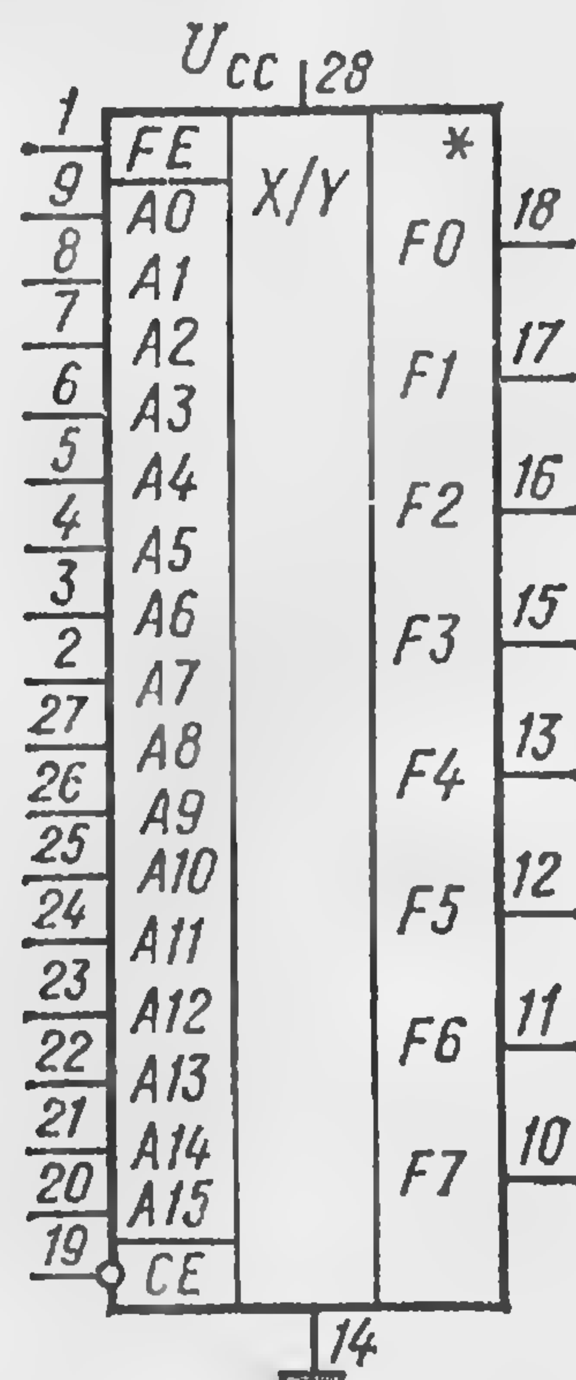


Рис. 1. Условное графическое обозначение ПЛМ К(Р)556РТ1/РТ2

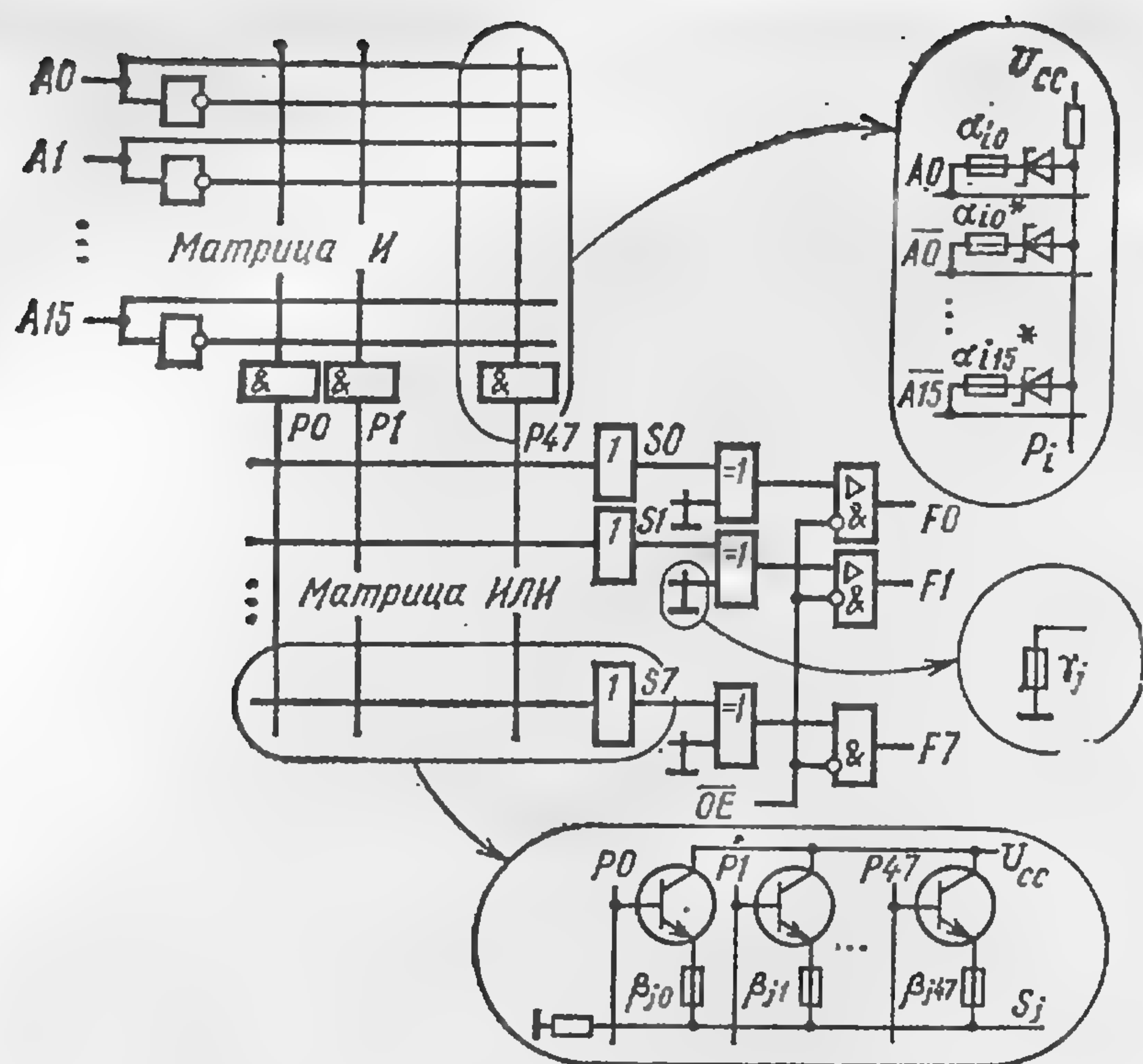


Рис. 2. Логическая схема ПЛМ

пускается в рабочем i-конъюнкторе ПЛМ оставлять одновременно обе парные перемычки ($\alpha_{im}=\alpha^*_{im}=1$) для неиспользуемого входа A_m, так как это влечет за собой тождественно нулевое значение P_i. Все перемычки резервных схем И, как правило, оставляют целыми.

Второй логический уровень образует матрица из восьми 48-входовых схем ИЛИ (дизъюнкторов), по одной на каждый выход ПЛМ:

$$S_j = \sum_{i=0}^{47} \beta_{ji} P_i,\ j=\overline{0,7}.$$

Плавкие перемычки β_{ji} позволяют организовать выборочную сумму произведений $P_{i,i}=\overline{0,47}$. В исходном состоянии, когда все перемычки целы ($\beta_{ji}=1$, $j=\overline{0,7}$, $i=\overline{0,47}$) и $P_i=0$, $i=\overline{0,47}$, все дизъюнкции $S_{j,j}=\overline{0,7}$, тождественно равны нулю. В рабочей ПЛМ допускается оставлять перемычки β_{ji}, $j=\overline{0,7}$, для каждого неиспользуемого (резервного) i-конъюнктора, если его выход равен нулю. Эта связь может в дальнейшем потребоваться для редакции рабочей ПЛМ, так же как и все другие резервные связи микросхемы.

На выходах схем ИЛИ находится слой программируемых инверторов, построенный на схемах 2-входного «исключающего ИЛИ», и ряд буферов, открываемый сигналом $\overline{OE}$:

$$F_j=\overline{OE}(\overline{\gamma_j 1 \oplus S_j}),\ j=\overline{0,7}.$$

Удалением перемычек γ_j, $j=\overline{0,7}$, можно выборочно изменить полярность выходов схем ИЛИ. В исходном состоянии ($\gamma_j=1$, $j=\overline{0,7}$) при выбранной схеме ($\overline{OE}=0$) на всех выходах считывается ноль. Выходные буферы выполняются по схеме с открытым коллектором или коллектором с тремя состояниями.

Процедуры программирования ПЛМ

Физические принципы программирования ПЛМ аналогичны принципам записи информации в ПЗУ [8]. В процессе эксплуатации ПЛМ могут возникнуть следующие задачи: анализ состояния микросхемы, начальное программирование «чистой» ПЛМ, повторное программирование (редактирование) логической матрицы. Выполнение этих задач основано на использовании ряда простейших процедур (примитивов) программирования ПЛМ:

прожиг связи матрицы И (Р1);
контроль связи матрицы И (V1);
прожиг связи матрицы ИЛИ (Р2);
контроль связи матрицы ИЛИ (V2);
прожиг связи слоя НЕ (Р3);
контроль связи слоя НЕ (V3).

Можно построить несколько алгоритмов, реализующих задачи анализа, программирования и редактирования, которые отличаются друг от друга различными последовательностями выполнения примитивов прожига и контроля. Эти возможности следует использовать для повышения эффективности систем программирования ПЛМ. Существуют некоторые варианты выполнения самих примитивов.

Для реализации процедур прожига и контроля в состав схемы ПЛМ введен ряд дополнительных элементов. Программирующий дешифратор и адресные формирователи (АФ) (рис. 3) работают только в режимах программирования и контроля. Адресные формирователи

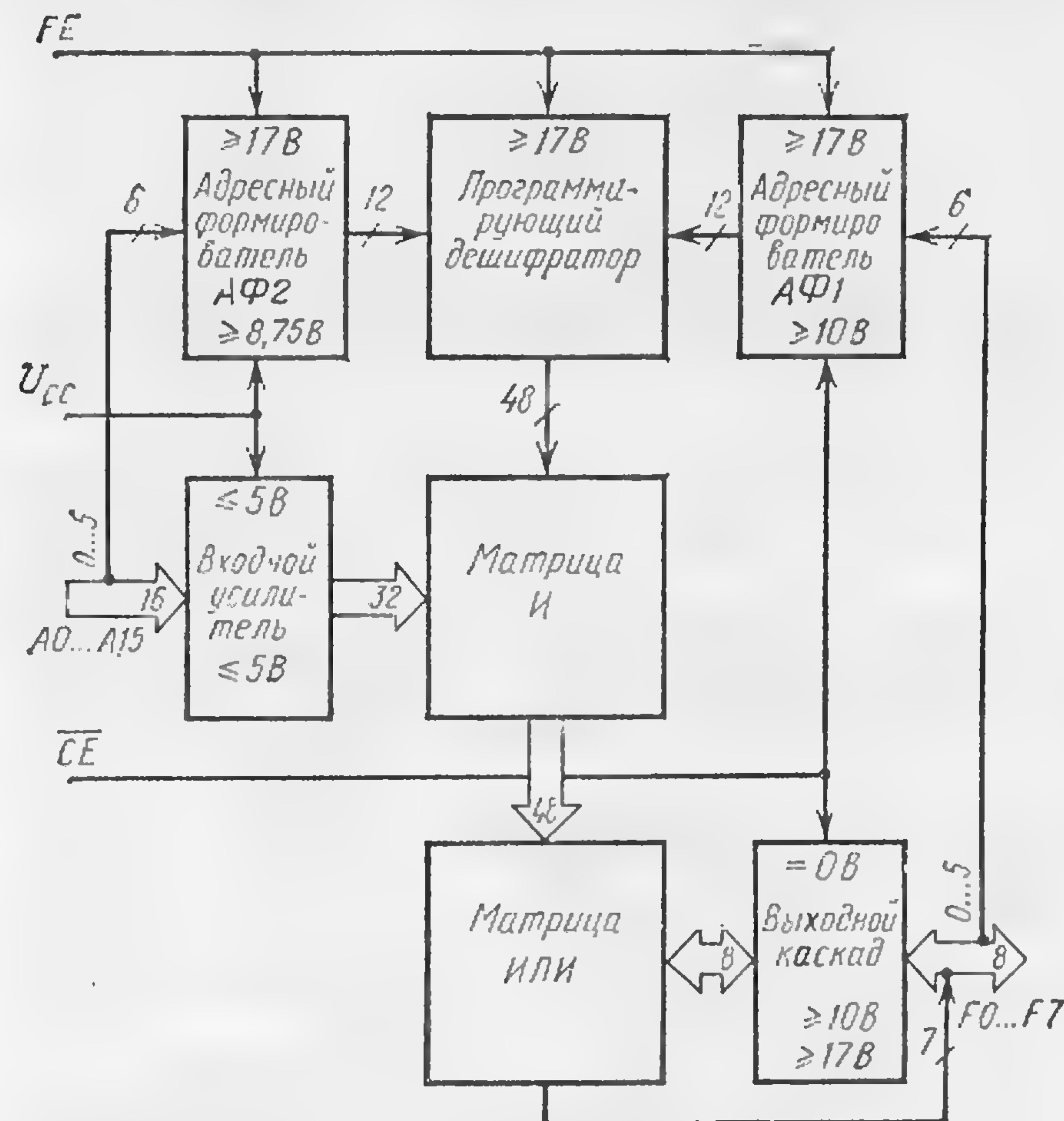


Рис. 3. Структурная схема ПЛМ

управляют дешифратором, определяя адрес выбранной конъюнкции P_i, $i=\overline{0,47}$. Формирователь АФ1 используется при работе с матрицей И, АФ2 — при программировании матрицы ИЛИ. В состав АФ и программирующего дешифратора входят также схемы, обеспечивающие их включение и выключение в зависимости от режима работы. Эти пороговые схемы управляются напряжением на входах U_{cc}, FE и $\overline{CE}$ микросхемы.

Выберем начальное состояние сигналов на внешних выводах прибора:

$$\begin{cases} GND = 0\ В, \\ U_{cc} = U_0, \\ FE = U_0, \\ \overline{CE} = U_1 \\ A0 \ldots A15 = U_1, \\ F0 \ldots F7 = 5\ В\ через\ R = 10\ кОм. \end{cases}$$

Здесь, как и далее, U_0, U_1 — стандартные для ТТЛ-приборов напряжения «Лог. 0» и «Лог. 1» соответственно.

Программирование и контроль матрицы И

Режим реализуется при напряжении питания $U_{cc}=5$ В и отключенных выходных каскадах ($\overline{CE}=0$ В). Если $\overline{CE}=10$ В, то открывается АФ1 и программирую-

щий дешифратор выбирает сборку И, указанную кодом F0...F5 (F0 — младший разряд кода). Для удаления требуемой перемычки выбранной сборки необходимо закрыть все выходы входных усилителей (прямые и инверсные), кроме программируемого. Для этого на входы всех усилителей, кроме одного, подается напряжение 10 В. На вход выбранного усилителя подают напряжение U_1, если необходимо пережечь перемычку α^*_{lm}, и U_0, если α_{lm}. Импульс программирующего тока формируется при подаче на вход FE напряжения 17 В.

В режиме контроля FE=0 В; при этом ток источника питания втекает через матрицу И при наличии проверяемой перемычки или через матрицу ИЛИ при ее отсутствии. Схема контроля матрицы И, связанная с выходом F7, фиксирует наличие или отсутствие тока в матрице ИЛИ.

Процедура P1 прожига связи матрицы И разрушает одну связь α_{lm} или α^*_{lm}.

P11. Повысить U_{cc} до 5 В±5 %, I≤1 А.

P12. 1. На F0...F5 подать двоичный адресный код i, адресующий выбранный конъюнктор.
2. На A0...A15 подать 10 В±10 %, I≤2 мА.
3. На вход A_m подать
U_1, если удаляется α^*_{lm} ·
U_0, если удаляется α_{lm}.

P13. На FE подать напряжение 17 В±1 В, I≤≤300 мА, время нарастания $\tau_ф$=5...25 мкс.

P14. На $\overline{CE}$ подать импульс амплитудой 10 В±±5 %, I≤5 мА, длительность которого $\tau_п$=25...100 мкс. По окончании импульса $\overline{CE}$ возвратить в исходное состояние.

P15. Возратить потенциал FE в исходное состояние.

P16. 1. На входе A_m восстановить напряжение 10 В±10 %.
2. Возвратить A0...A15 в исходное состояние.
3. Снять адресный код с F0...F5.

P17. Снизить U_{cc} до исходного значения.

Интервал времени между двумя соседними операциями должен быть более 1 мкс.

При использовании непрерывной серии попыток прожига ряд операций может быть сокращен. Это касается граничных пар P11, P12 и P16, P17, которые включаются только в первом и последнем циклах серии.

Процедура V_1 контроля связи матрицы И используется для анализа состояния отдельной пары перемычек α_{lm} и α^*_{lm} ПЛМ. Процедура похожа на примитив P1 прожига связи, однако в ней отсутствует импульс FE, формируемый операциями P13 и P15.

V11. Повысить U_{cc} до 5 В±5 %, I≤1 А.

V12. 1. На F0...F5 подать двоичный адресный код i, адресующий выбранный конъюнктор.
2. На A0...A15 подать 10 В±10 %, I≤2 мА.

V13. На $\overline{CE}$ подать напряжение 10 В±5 %, I≤≤5 мА.

V14. На входе A_m снизить напряжение до U_1. Зафиксировать значение логической переменной на выходе F7.

V15. На входе A_m снизить напряжение до U_0. Зафиксировать значение на выходе F7.

V16. На входе A_m восстановить напряжение 10 В±5 %.

V17. Снизить напряжение на $\overline{CE}$ до исходного U_1.

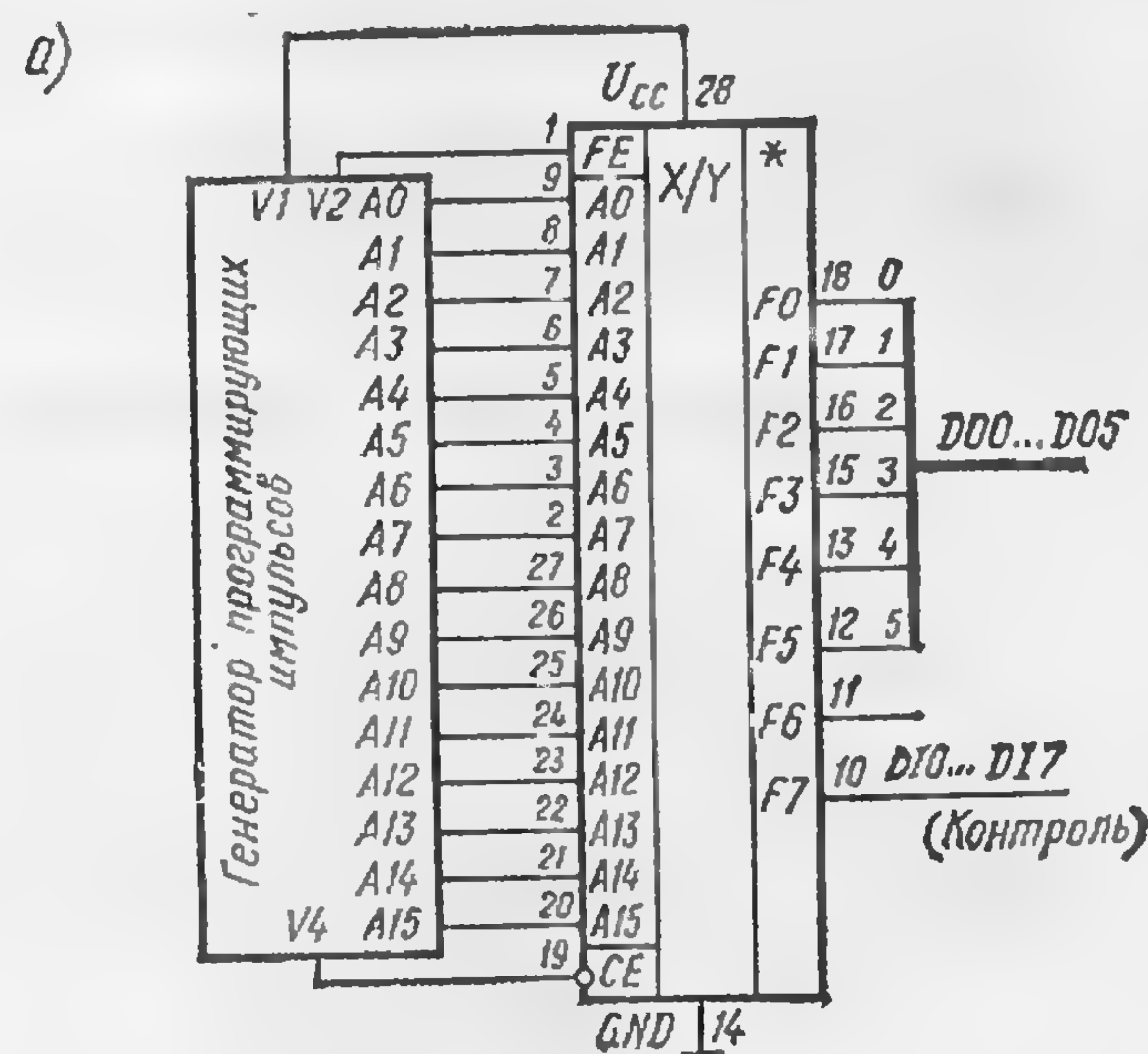
V18. 1. Возвратить A0...A15 в исходное состояние.
2. Снять адресный код с F0...F5.

V19. Снять напряжение с U_{cc}.

О состоянии перемычек судят по итогам наблюдения выхода F7:

$F7=A_m$, $\alpha_{lm}=1$ и $\alpha^*_{lm}=0$,
$F7=\overline{A}_m$, $\alpha_{lm}=0$ и $\alpha^*_{lm}=1$,
$F7=1$, $\alpha_{lm}=0$ и $\alpha^*_{lm}=0$,
$F7=0$, $\alpha_{lm}=1$ и $\alpha^*_{lm}=1$.

При выполнении процедуры контроля непосредственно после процедуры прожига операции P16 и P17 в процедуре прожига и V11 и V12 в процедуре контроля можно исключить. Временные диаграммы выполнения серии таких совмещенных процедур прожига и контроля PV1 приведены на рис. 4. При непрерывном контроле массива перемычек матрицы И некоторые операции можно опустить (контроль выбранной схемы И требует цикла, состоящего лишь из операций V14... ...V16).



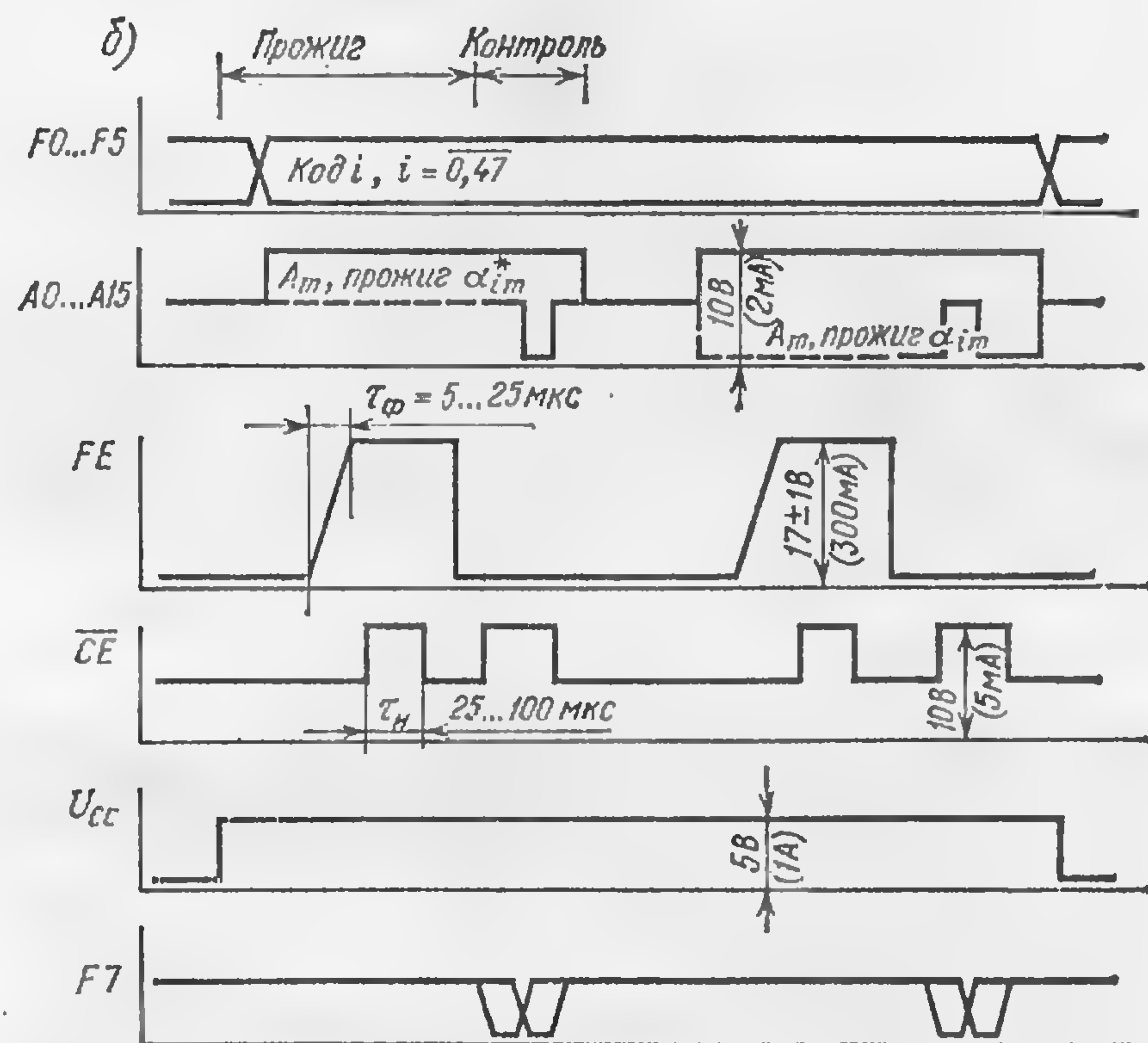


Рис. 4. Программирование матрицы И: схема подключения (а), временные диаграммы (б)

Программирование и контроль матрицы ИЛИ

Режим осуществляется при напряжении питания U_{cc}=8,75 В, которое разрешает работу АФ2. На входы A0...A5 подается код (A0 — младший разряд кода), соответствующий номеру выбранного конъюнктора. На выход функции, из которой исключается выбранная конъюнкция, — напряжение 10 В. Импульс программирующего тока, протекающий по адресованной таким способом перемычке, формируется при подаче на программирующий вход FE напряжения 17 В, а на вход $\overline{CE}$ — 10 В. Контроль записанной в матрицу ИЛИ информации выполняется аналогично, только при напряжении на входах FE и $\overline{CE}$, равном U_0. О наличии или отсутствии проверяемой перемычки судят по уровню сигнала на выходе микросхемы.

Процедура P2 прожига связи матрицы ИЛИ удаляет выделенную связь β_{ji} из матрицы суммирования.

P21. На A0...A5 подать двоичный код i, адресующий терм P_i.

P22. U_{cc} повысить до 8,75±0,25 В, $I \leqslant 1$ А.

P23. На программирующем выходе F_j напряжение поднять до 10 В±5 %.

P24. На FE-входе напряжение повысить до 17±1 В, $I \leqslant 300$ мА, длительность фронта $\tau_ф = 5...25$ мкс.

P25. На вход $\overline{CE}$ подать импульс амплитудой 10±0,5 В, $I \leqslant 5$ мА, длительность импульса $\tau_и = 25...100$ мкс.

P26. Вернуть FE в исходное состояние U_0.

P27. Снять напряжение 10 В с выхода F_j.

P28. Снять напряжение с U_{cc}.

Процедура V2 проверки целостности перемычек матрицы ИЛИ.

V21. На A0...A5 подать двоичный код i, адресующий вход схем ИЛИ.

V22. U_{cc} повысить до 8,75±0,25 В, $I \leqslant 1$ А.

V23. На вход $\overline{CE}$ подать напряжение U_0.

V24. О состоянии перемычки β_{ji} судят по значению логической переменной на выходе F_j:

$$F_j=0, \quad \beta_{ji}=0 \text{ при } \gamma_j=1,$$
$$\beta_{ji}=1 \text{ при } \gamma_j=0,$$
$$F_j=1, \quad \beta_{ji}=1 \text{ при } \gamma_j=1,$$
$$\beta_{ji}=0 \text{ при } \gamma_j=0.$$

V25. Вернуть $\overline{CE}$ в исходное состояние U_1.

V26. Снять напряжение с выхода U_{cc}.

При выполнении серий, состоящих из процедур P2 и V2, справедливы все замечания, которые сделаны для процедуры программирования матрицы И. Временные диаграммы серии процедур PV2 (прожиг и следующий за ним контроль) для некоторой перемычки матрицы ИЛИ приведены на рис. 5.

Программирование и контроль слоя НЕ

Пережигание перемычки γ_j происходит при подаче на выход F_j напряжения 17 В. При этом срабатывает схема программирования перемычки в выходном каскаде и через прожигаемую перемычку проходит разрушающий ее ток.

При контроле состояния перемычки на схему подают повышенное напряжение питания U_{cc}=8,75 В, а на адресные входы A0...A5 код 111111B в ТТЛ-уровнях. При этом ни одна из 48 конъюнкций не выбирается и, следовательно, ток в матрицу ИЛИ не поступает. По состоянию выхода F_j можно судить о целостности перемычки γ_j.

Процедура P3 прожига перемычки в слое НЕ состоит из одной операции.

P31. На программирующий выход F_j подать импульс напряжения 17±1 В. Время нарастания $\tau_ф = 5...25$ мкс. Длительность импульса $\tau_и = 25...100$ мкс.

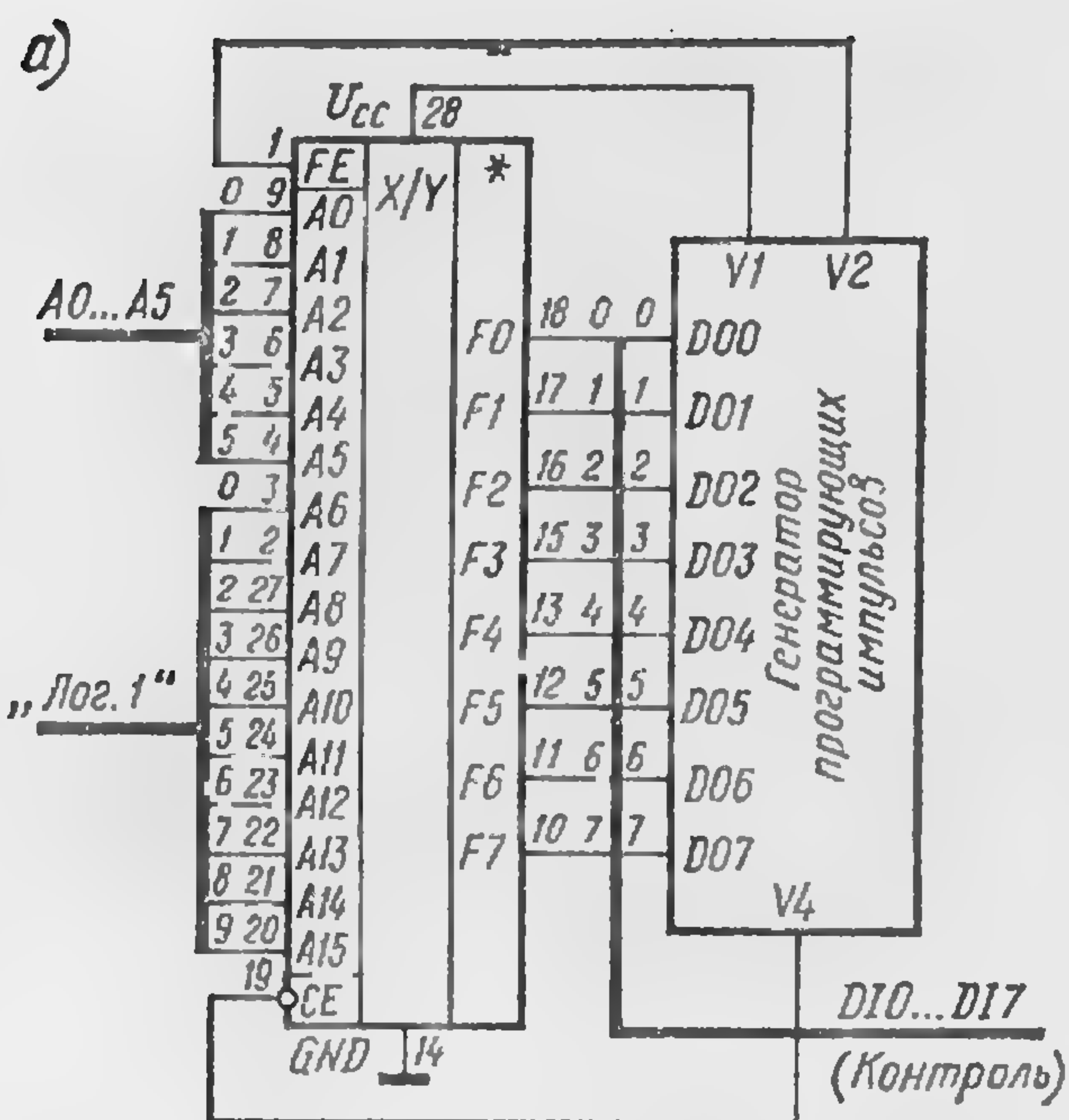
Процедура V3 контроля состояния перемычки определяется следующей последовательностью операций:

V31. U_{cc} повысить до 8,75±0,25 В, $I \leqslant 1$ А.

V32. На вывод $\overline{CE}$ подать напряжение U_0.

V33. Контроль состояния перемычки γ_j выполняется по состоянию выхода F_j:

$$F_j=1, \quad \gamma_j=0,$$
$$F_j=0, \quad \gamma_j=1.$$



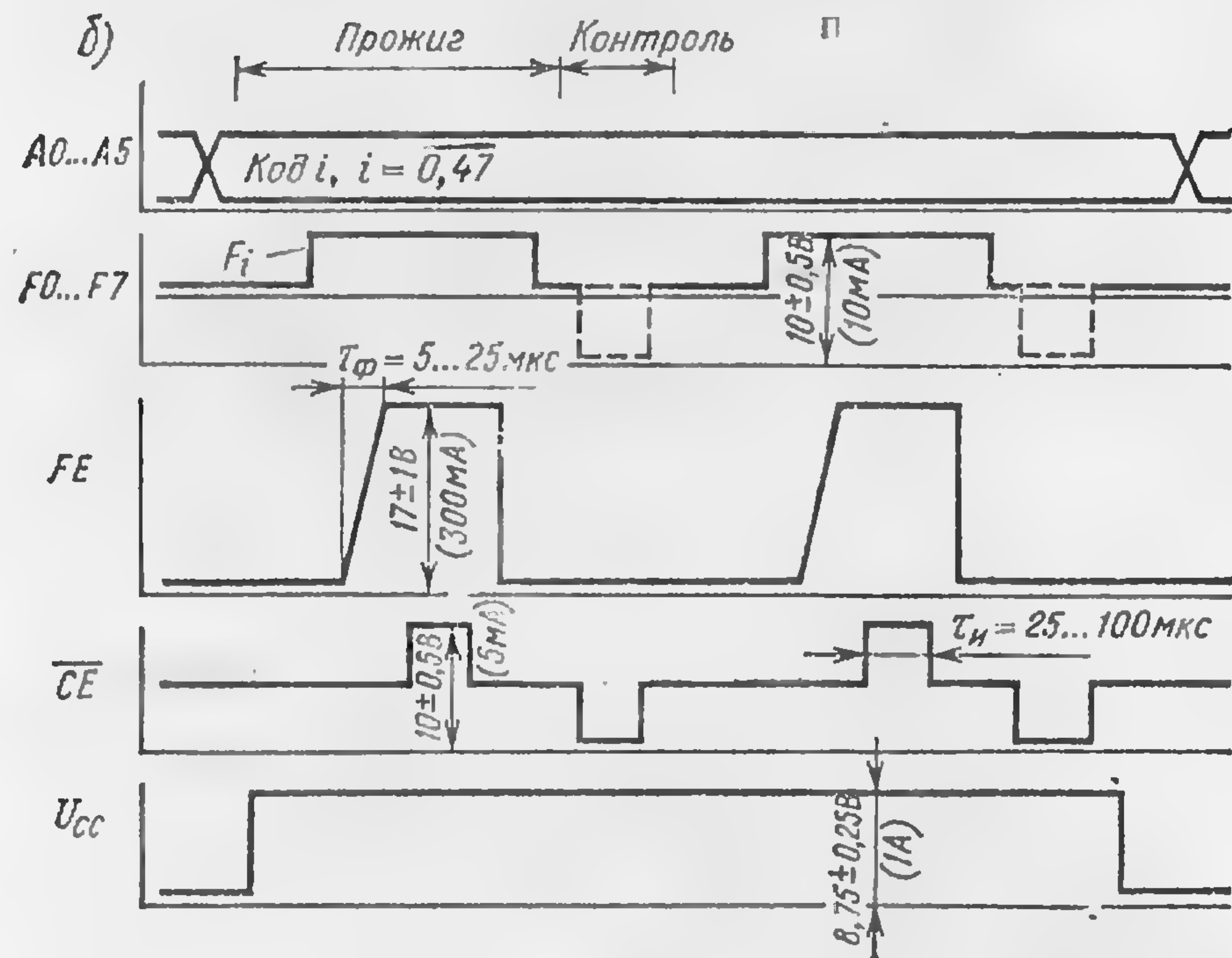


Рис. 5. Программирование матрицы ИЛИ: схема подключения (*а*), временные диаграммы (*б*)

V34. Поднять напряжение на $\overline{CE}$ до U_1.

V35. Снять U_{cc}.

Серия совмещенных процедур PV3 прожига и контроля перемычки γ_j приведена на рис. 6.

Программирование ПЛМ

В исходном незапрограммированном состоянии все перемычки ПЛМ целы, поэтому

$$P_i=0, \qquad i=\overline{0,47},$$
$$S_j=0, \qquad j=\overline{0,7},$$
$$F_j=0, \qquad j=\overline{0,7}.$$

Программирование микросхемы заключается в выборочном разрушении связей α_{im}, α^*_{im}, β_{ji} и γ_j, $i=\overline{0,47}$, $m=\overline{0,15}$, $j=\overline{0,7}$, импульсами тока длительностью 25 мкс$\leqslant \tau \leqslant$100 мкс. Ненужные перемычки удаляются последовательно одна за другой. Процедура разрушения очередной связи состоит из двух серий прожигающих импульсов: основной и дополнительной (рис. 7). После каждого прожигающего импульса предусматривается промежуточный контроль разрушения связи, что эквивалентно использованию $PV_к$-процедур. Процедура разрушения начинается с подачи основной серии, включающей N1=1000...4000 импульсов прожига. После первого удачного контроля основная серия обычно прерывается и начинается прохождение дополнительной серии, состоящей из N2=40...100 импульсов. При первом об-

наружении восстановления перемычки во второй серии весь алгоритм следует повторить.

Обычно число повторов ограничивают целым значением L. Процедура разрушения очередной связи завершается удачно только при полном прохождении всей дополнительной серии. После этого можно перейти к удалению следующей перемычки.

Последовательное удаление связей в «чистой» ПЛМ удобно разбить на три этапа:

программирование уровня активности выходов (прожиг необходимых связей γ_j, $j=\overline{0,7}$);

программирование матрицы И (избирательное удаление перемычек α_{im} и α^*_{im}, $i=\overline{0,47}$, $m=\overline{0,15}$);

программирование матрицы ИЛИ (расплавление ненужных связей β_{ji}, $i=\overline{0,47}$, $j=\overline{0,7}$).

Алгоритмы удаления перемычек внутри этапа отличаются друг от друга операциями прожига $P_к$ и контроля $V_к$, входящими в основную и дополнительную серии обобщенной процедуры прожига.

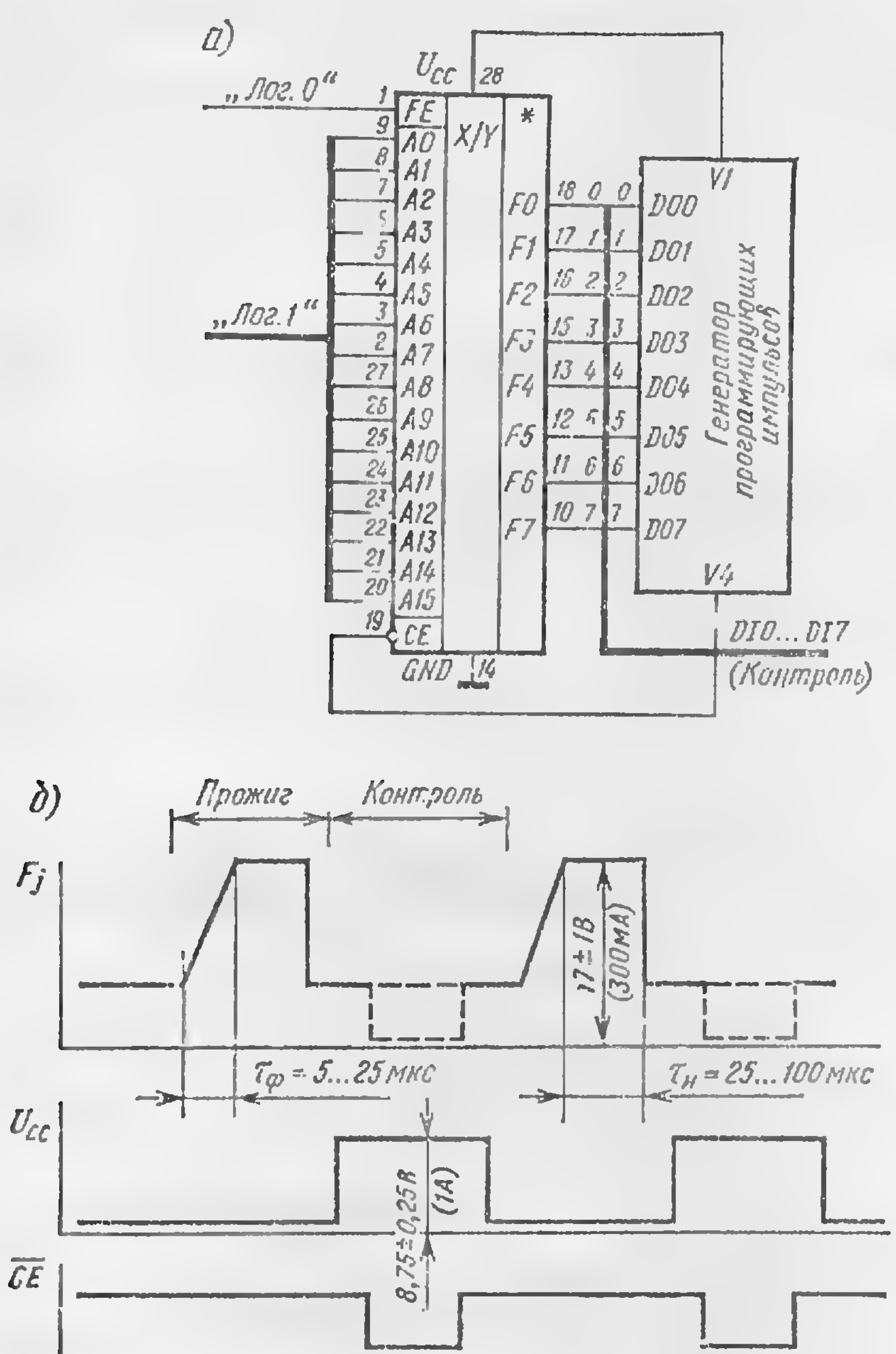


Рис. 6. Программирование слоя НЕ: схема подключения (а), временные диаграммы (б)

Следует отметить, что существуют и другие алгоритмы программирования ПЛМ. От рассмотренного они отличаются тем, что контроль состояния перемычки выполняется не после каждой попытки прожига, а через несколько попыток. Внутри серии отдельная попытка может относиться к одной или разным связям. После проверки состояния серия повторяется по отношению ко всем или только к неразрушенным перемычкам.

Анализ состояния микросхемы

Задача анализа состояния ПЛМ возникает при начальном контроле БИС на незапрограммированность или наличие резервных связей, изучении функции неизвестной запрограммированной ПЛМ, итоговом контроле, завершающем процесс программирования или редактирования. Во всех случаях выполняется анализ состояний

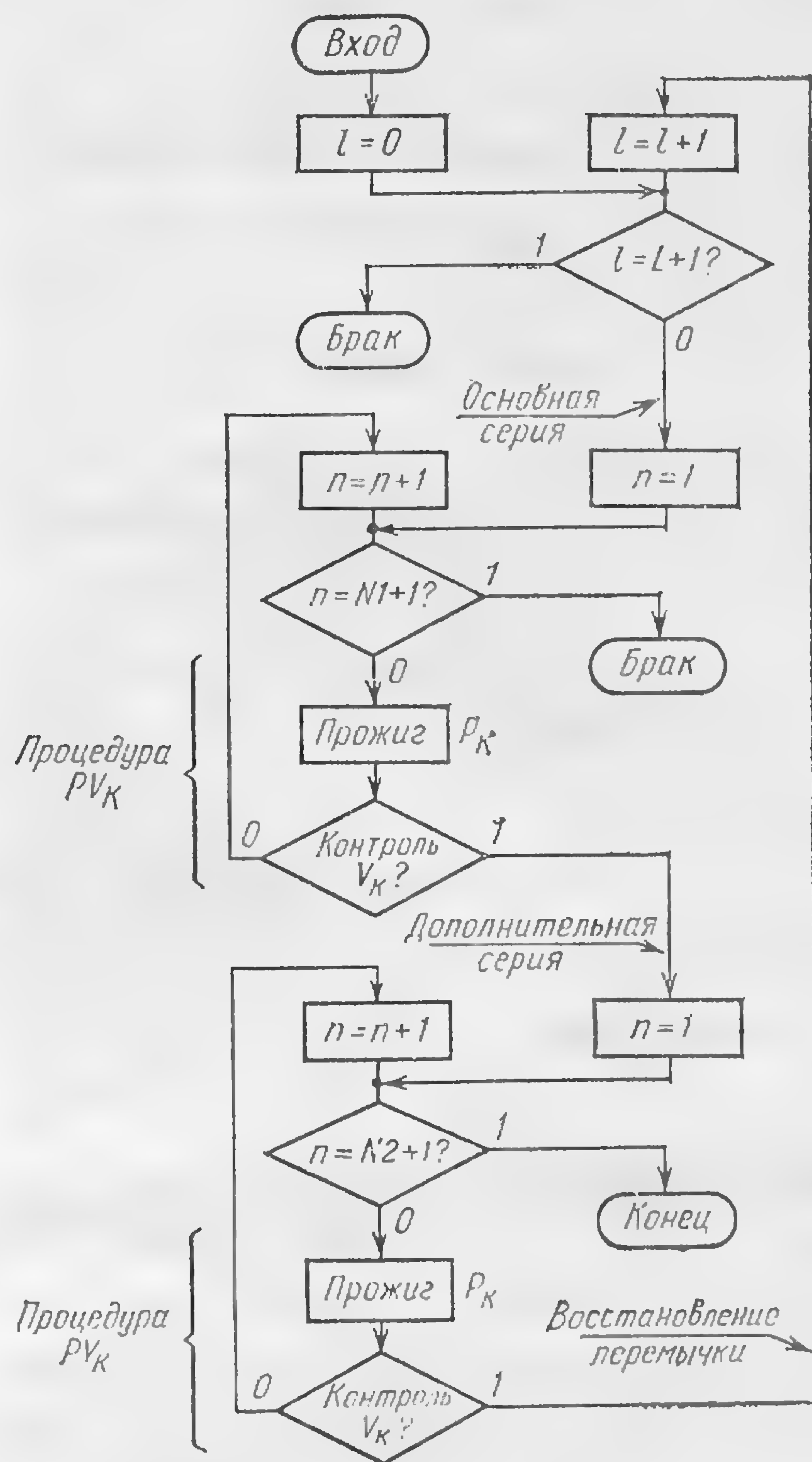


Рис. 7. Обобщенная процедура прожига перемычки

матрицы И, слоя НЕ, матрицы ИЛИ (рис. 8). Каждый этап состоит из серии соответствующих примитивных операций контроля, рассмотренных выше.

Анализ матрицы И дает возможность проверить состояние входов конкретного конъюнктора, вхождение некоторой переменной во все конъюнкторы, выписать весь набор конъюнкций, реализованных в ПЛМ. Анализ состояния слоя НЕ дает информацию об уровне активности выходного сигнала. При контроле матрицы ИЛИ проверяется вхождение конкретного конъюнктора в состав функций, определяется набор всех конъюнкций, включенных в некоторую функцию, составляются все реализованные в ПЛМ дизъюнкции. В конечном итоге определяется карта состояния ПЛМ и выполняемая ею функция.

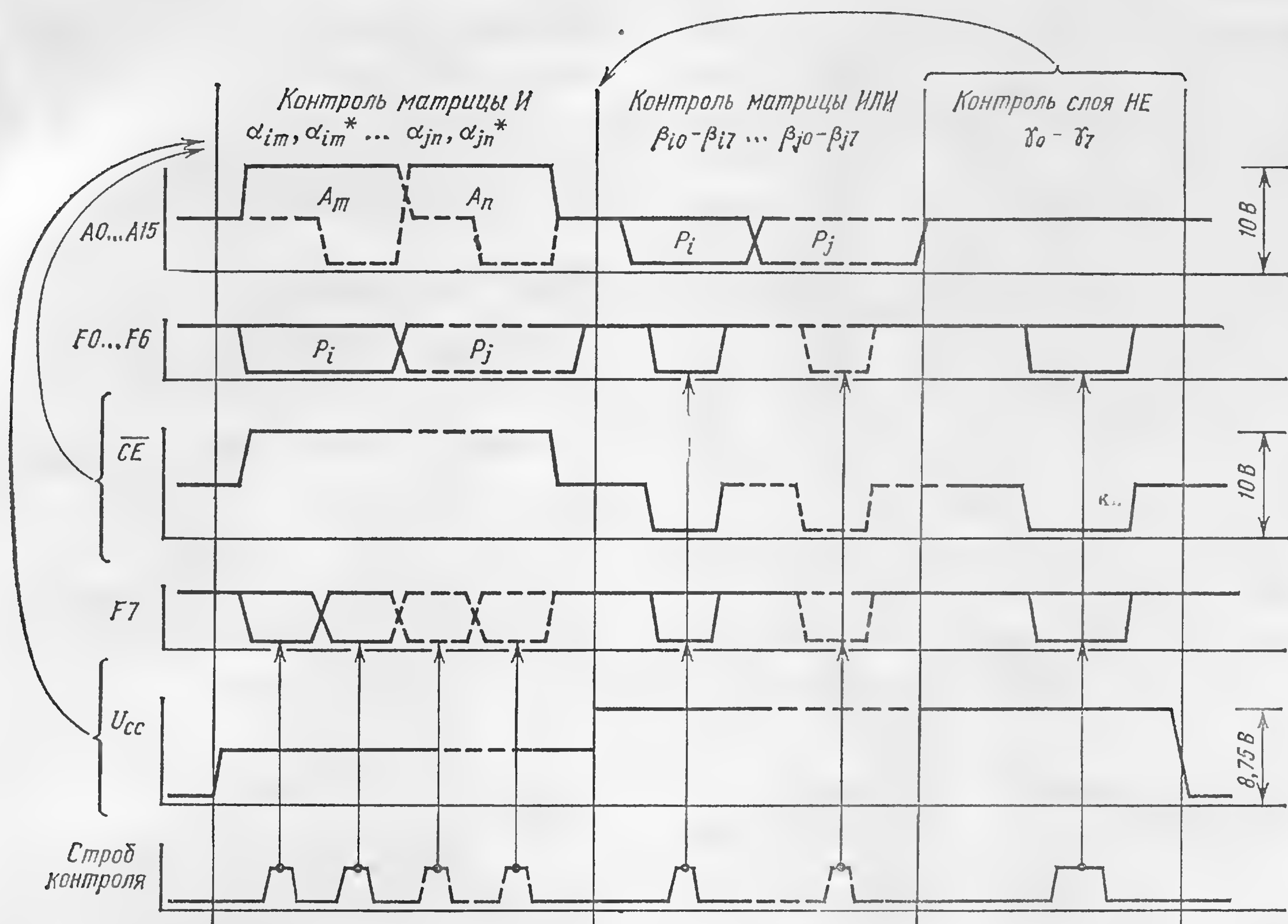


Рис. 8. Анализ состояния ПЛМ

Редактирование ПЛМ

Редактирование возможно только с помощью добавочного удаления неразрушенных связей. Это обстоятельство ограничивает возможности редактирования ПЛМ, которые состоят в следующем:

замена высокого уровня активности на низкий (прожиг γ_j);

исключение конъюнкции из выходной функции (прожиг β_{ji});

исключение входной переменной из конъюнктора (прожиг α_{im} или α^*_{im});

добавление нового конъюнктора в выходную функцию из числа резервных (прожиг α_{im}, α^*_{im}, β_{ji});

добавление новой функции из числа резервных с использованием как старых, так и новых конъюнкторов

Алгоритмы редактирования строятся из примитивов по аналогии с процессом программирования прибора.

ЛИТЕРАТУРА

1. Электроника, 1979, № 14, с. 24—32.
2. Электроника, 1984, № 16, с. 91, 92.
3. Электроника, 1984, № 15, с. 3—5.
4. Электроника, 1984, № 19, с. 5, 6.
5. Электроника, 1984 № 21, с. 93.
6. Электроника, 1984, № 9, с. 5, 6.
7. ОСТ 11 340.915—82.
8. Щербаков О. А. Физические основы записи информации в ПЗУ. — Микропроцессорные средства и системы, 1985, № 3, с. 72—75.

Статья поступила 17 июля 1985 г.

ДЕЙСТВУЕТ СЕМИНАР

На базе Института общей физики АН СССР работает ежемесячный (каждый второй четверг) семинар «Программное обеспечение систем автоматизации реального времени».

Семинар имеет следующие проблемные секции:

операционные системы реального времени;
алгоритмические языки и трансляторы;
библиотеки для научно-технических расчетов;
автоматизированное управление экспериментом;
машинная графика;
базы данных и информационно-поисковые системы;
обработка текста;
кросс-системы программирования;
распределенные управляющие системы и локальные сети ЭВМ.

Семинар объединяет разработчиков и пользователей систем автоматизации на базе ЭВМ, программно совместимых с линией СМ-4, и способствует координации работ по созданию ПО и обмену опытом и информацией о разработанном ПО. Участником семинара может быть любой специалист, интересы которого связаны с тематикой семинара.

Адрес для переписки: *Москва, 117923, ул. Вавилова, 38, ИОФ АН СССР. Телефон для справок: 135-03-97, Власова Лариса Николаевна.*

УДК 681.326.3

А. П. Дианов, Н. Н. Щелкунов

ТЕХНИЧЕСКИЕ СРЕДСТВА ПРОГРАММИРОВАНИЯ ЛОГИЧЕСКИХ СХЕМ

Программирование матричной логики аналогично методам записи информации в программируемые ПЗУ [1]. Это позволяет создавать единые программно-аппаратные средства для занесения информации в БИС матричной логики и ПЗУ, состоящие из специальных аппаратных модулей программирования и пакетов прикладных программ [2, 3].

Модуль программирования содержит ряд регистров, буферов, ключей и программируемых источников питания (ПИП), которые организуют сопряжение программируемой БИС с микропроцессорной системой. В его состав входят также аппаратные формирователи временных последовательностей, если эти последовательности не могут быть получены на программном уровне.

Процессы разработки дополнительных технических средств и специального программного обеспечения можно разделить и выполнять независимо, если предварительно определить программную модель программатора. Последняя представляет собой набор программно-доступных регистров модуля, их функциональное описание и способы доступа (рис. 1). В основу модели положена универсальная байтовая организация ввода-вывода. Модель содержит двунаправленный порт данных DB и два выходных порта адресной информации AL, AH. Это позволяет программировать микросхемы, имеющие 8 информационных линий и 16 адресных входов.

Адреса портов	Организация ввода-вывода	Функциональное назначение
5BH	7 DB 0	Ввод-вывод данных D0...D7
5CH	7 AL 0	Младший байт адреса A0...A7
5DH	7 AH 0	Старший байт адреса A8...A15
5EH	[illegible] 3 AS 0	Управление буферами адреса-данных
5FH	7 6 5 4 3 0	Управляющее слово ППА1
68H	E3	Амплитуда D0...D7
69H	E4	Амплитуда выборки кристалла
6AH	VS 1 SS	[illegible] включением [illegible] ПИП
6BH	7 4 3 0	Управляющее слово ППА2
6CH	7 E1 0	Амплитуда U_{cc}
6DH	7 E2 0	Амплитуда напряжения программирования
6EH	7 E5 0	Амплитуда A0...A15
6FH		Управляющее слово ППА3

Рис. 1. Программная модель модуля программатора

Основу двунаправленного порта данных DB составляют два многорежимных буферных регистра К589ИР12, имеющих адрес 5BH. Остальные девять портов модуля разбиты на три группы по три в каждой (рис. 1). Отдельная группа реализована на одном программируемом периферийном адаптере (ППА) типа КР580ВВ55. По этой причине в каждой группе из трех портов появляется четвертый порт управляющего слова, который используется для программирования режима работы ППА. Все адаптеры программируются для работы в режиме прямого вывода по всем трем каналам.

Структурная схема модуля программатора аналогична предложенной в работе [2]. Модуль построен на базе внутримодульной магистрали типа Microbus, работающей в синхронном режиме. Интерфейсная логика модуля организует стык этой магистрали с системной магистралью типа Multibus, используемой микроЭВМ. Интерфейсная логика включает шинный драйвер, логику выборки кристаллов, формирующую сигналы $\overline{CS0}$ (обращение к порту данных), $\overline{CS1}$ (выбор ППА1), $\overline{CS2}$ (выбор ППА2) и $\overline{CS3}$ (обращение к ППА3), буферные схемы управляющих сигналов $\overline{I/OR}$, $\overline{I/OW}$ и адресов A0, A1, а также логику подтверждения обмена, генерирующую сигнал $\overline{XACK}$. Логика выборки кристаллов и буферы A0, A1 реализованы на микросхеме КР556РТ5. При выборе общесистемной магистрали, отличной от Multibus, интерфейсная логика должна быть заменена.

В состав модуля программатора входят пять байтовых портов E1...E5, которые служат для программирования источников питания, стандартизации и упрощения реализации процессов программирования ППЗУ и ПЛМ. Выход первого источника резервируется под питание программируемых БИС. Второй источник необходим для формирования напряжения программирования Vpp или строба FE при записи информации в УФРПЗУ или ПЛМ соответственно. Напряжение третьего источника определяет амплитуду кода D0...D7, выданного через двунаправленный порт данных DB. Четвертый — управляет напряжением на входе выборки кристалла программируемой БИС, что важно для ППЗУ и ПЛМ. Пятый — резервируется для задания амплитуды адресного кода A0...A15, необходимого при программировании ПЛМ. Диапазон регулировки выходных напряжений источников лежит в пределах 0...25,6 В. Шаг изменения напряжения равен 0,1 В.

Блок программируемых источников питания строится так, как предложено в работе [2]. Каждый источник содержит 8-разрядный буферный регистр, реализованный на базе отдельного порта ППА, ЦАП К572ПА1, быстродействующий операционный усилитель К544УД2 и выходные транзисторы.

Кроме этого модуль программатора включает два программно-доступных порта управления CW1 (адрес 5EH), CW2 (адрес 6AH). В одном из них CW1 выделено 4-разрядное поле AS, которое кодирует номер адресной линии Am, участвующей в адресации плавкой перемычки матрицы И. При этом состояние соответствующего разряда из адресных портов AL, AH будет указывать на одну из двух перемычек, связанных с ней (см. рис. 1). «Лог. 0» в разряде адресует перемычку, связанную с m-входом через прямой буфер, а «Лог. 1» — перемычку, связанную с m-входом через инвертирующий буфер.

Биты $\overline{EN5}$ EAS порта CW1 используются для управления напряжением на адресных линиях A0...A15. Бит $\overline{EN5}$ резервируется под инверсный сигнал разрешения подачи на адресные входы программируемой БИС напряжения E5·0,1 В независимо от состояния портов AL, AH. Когда $\overline{EN5}=1$, на них должен присутствовать логический код состояния адресных портов в стандартных ТТЛ-уровнях. Второй бит EAS разрешает выбор адресной линии по коду поля AS. При этом на выбранную m-линию адреса независимо от сигнала $\overline{EN5}$ передается состояние соответствующего разряда из портов AL, AH в ТТЛ-уровнях. Совместное действие разрядов $\overline{EN5}$ и EAS иллюстрируется табли-

Управление адресными буферами

$\overline{EN5}$	EAS	Значение A0 ... A15, кроме A_m	Значение $A_{m,m}$-AS
1	X	Код AL, AH в ТТЛ-уровнях	m-бит AL, AH в ТТЛ-уровнях
0	0	E5·0,1 В	E5·0,1 В
0	1	E5·0,1 В	m-бит AL, AH в ТТЛ-уровнях

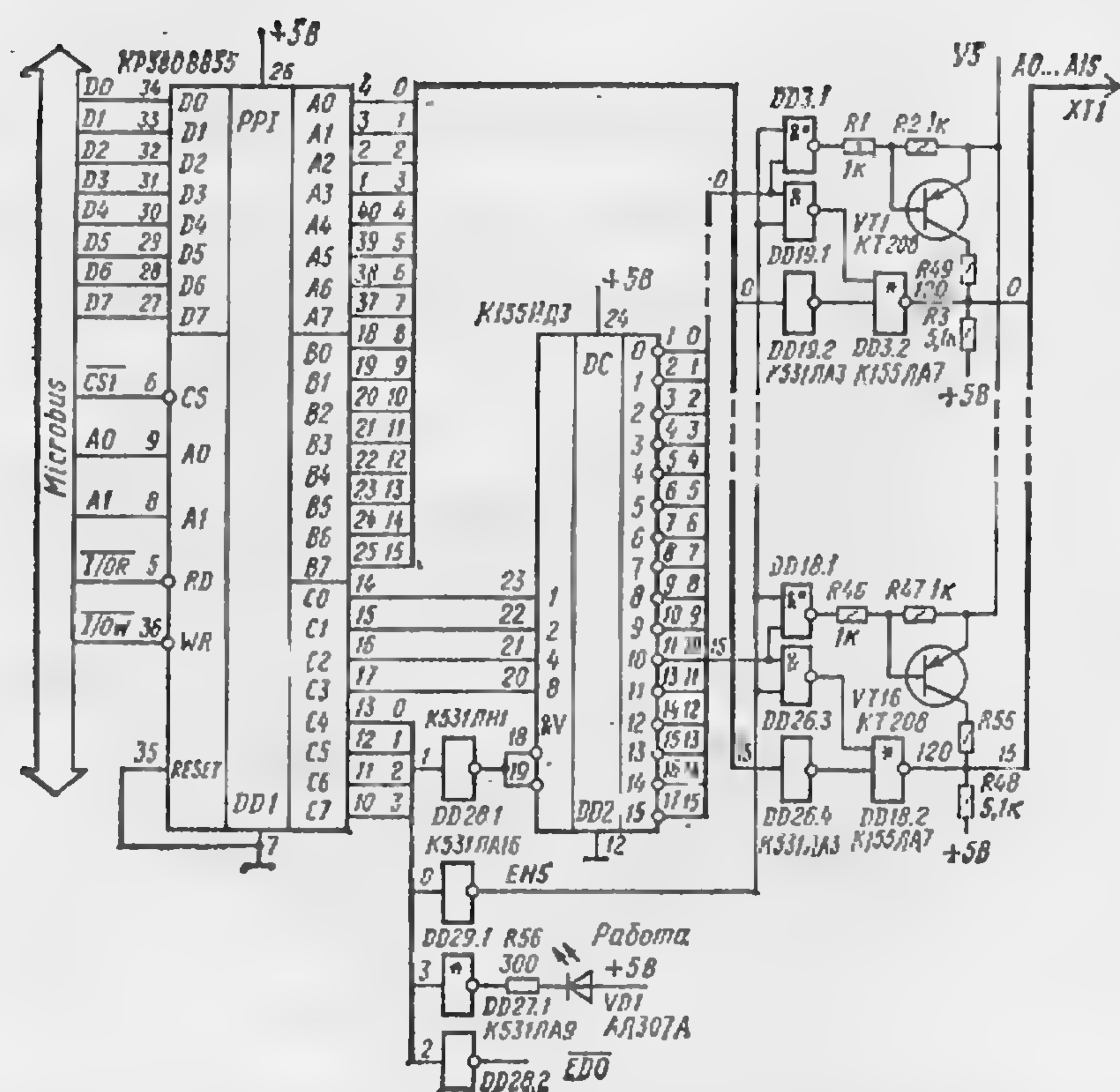


Рис. 2. Буферные регистры адреса

цей. Схема буферных регистров адреса приведена на рис. 2. Они содержат ППА КР580ВВ55 (DD1), дешифратор К155ИД3 (DD2) и 16 буферных ключей.

Схема буферных регистров данных представлена на рис. 3. При работе с ПЛМ 8-разрядные шины D1 и D0 соединяются на панельке программируемой БИС. В остальных случаях логика подключения приборов к шине данных аналогична описаниям, приведенным в работе [2]. Через разряд ED0 порта управления выводится сигнал управления выходными буферами. При ED0=0 на шине данных прибора присутствует код, выданный через порт DB с амплитудой V3=E3·01В. Если ED0=1, то «Лог. 0» и «Лог. 1» из регистра данных соответствуют высокоимпедансное состояние и уровень V3 на выходе буфера данных.

Работа выходных буферов определяется также величиной напряжения V3. При V3⩽5 В диоды VD1...VD8 смещаются в обратном направлении и, следовательно, отключают транзисторы VT1...VT8 от выходной шины. Если V3>5 В, то «Лог. 1» на выходе регистра данных DB соответствует уровень V3—$V3_0$ на шине D00...D07, где $V3_0 \simeq 1$ В — падение напряжения на насыщенном транзисторе и открытом диоде. При «Лог. 0» на выходе DB транзисторы VT1...VT7 закрываются. Влияние величины V3 на состояние буферов данных определяет их дополнительную зависимость от логики управления ПИП.

В режиме программирования ПЗУ ED0=0 и на шину данных выдается код DB с амплитудой V3>5 В. При ED0=1 и V3=0 В выходы D00...D07 находятся в высокоимпедансном состоянии, что дает возможность выполнить чтение программируемой БИС.

В режиме программирования матрицы ИЛИ ED0=1, а V3=10 В. При этом микросхемы DD3.2...DD10.2 находятся в высокоимпедансном состоянии и уровням «Лог. 0» и «Лог. 1» регистра данных соответствуют напряжение 5 В и 10 В на шине D00...D07.

В режиме программирования и контроля матрицы И ED0=0, V3=0 В. Шесть младших разрядов выходного регистра DD2 используются под адресацию конъюнктора, тогда как старший разряд всегда должен быть установлен в единицу. На выходах D00...D06 присутствует код конъюнктора в ТТЛ-уровнях, а буфер D07 находится в высокоимпедансном состоянии, освобождая старшую линию для считывания контрольной информации из ПЛМ.

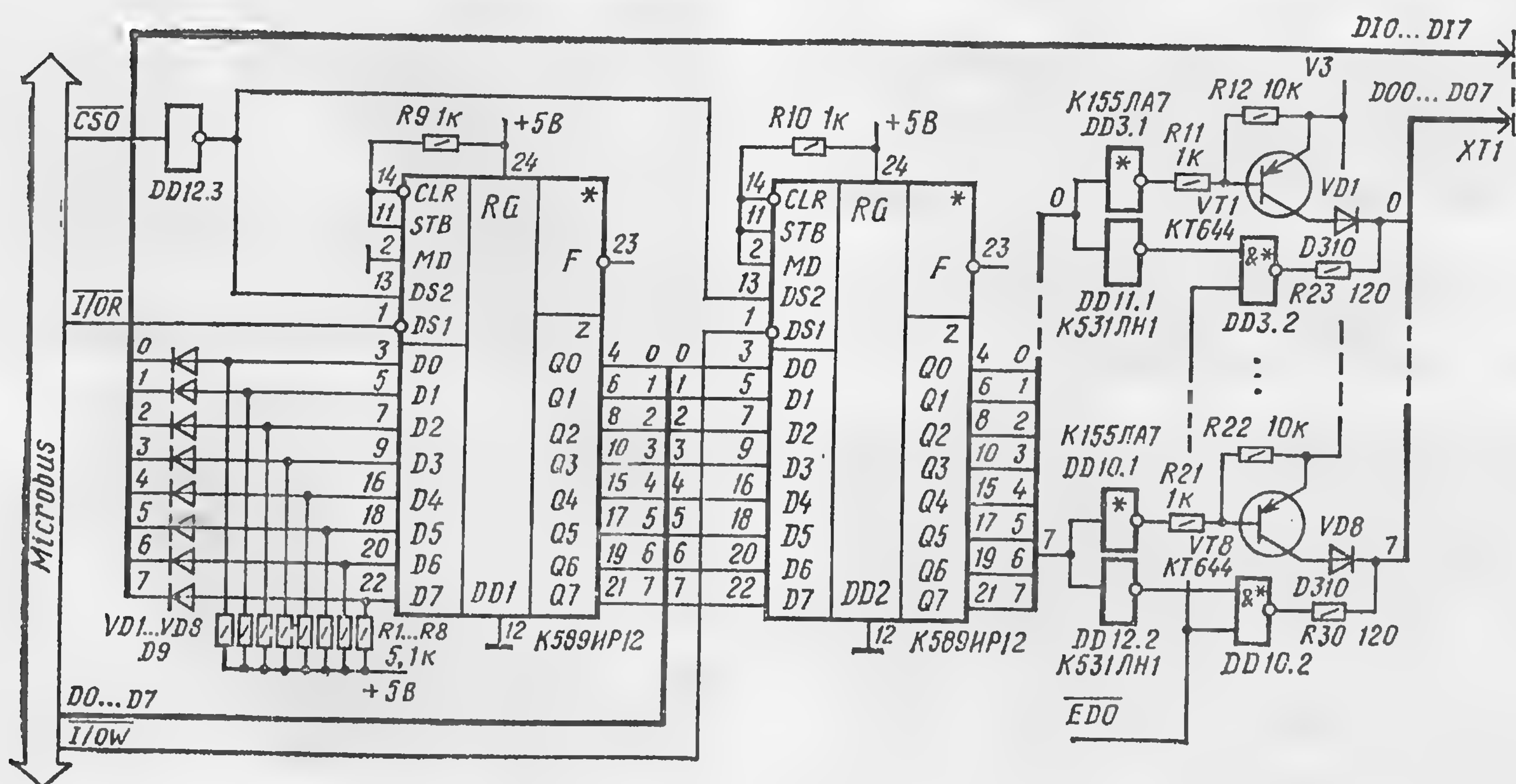


Рис. 3. Буферные регистры данных

Бит режима работы WM служит индикатором состояния программатора. В периоды, когда он не реализует никаких операций с БИС ПЗУ и ПЛМ, WM=0. В эти моменты разрешается смена программируемого прибора.

Порт CW2 используется для управления процессами включения и выключения ПИП. Управляя загрузкой портов данных E1...E5, можно получать необходимые временные диаграммы переключения ПИП. При этом минимальный временно́й интервал определяется типом используемой микроЭВМ. Для систем на базе микропроцессора КР580ВМ80А он составляет около 10 мкс. В тех случаях, когда программный код микроЭВМ не обеспечивает требуемого быстродействия переключения E1...E5, эти последовательности должны быть реализованы на аппаратном уровне.

В программной модели предусматривается 4-битное поле SS выбора одной из 16 последовательностей включения-выключения источников питания. С целью согласования аппаратуры с универсальной программой прожига версии 3.7 UPP рекомендуется последовательности включения запускать при каждой команде вывода информации в порт DB, следующей непосредственно за командой чтения. Последовательности выключения необходимо отрабатывать по каждой первой команде чтения порта данных, следующей за командой записи. Особенности запуска формирователя временных диаграмм могут потребовать дополнительного обращения к порту данных при считывании информации из программируемой БИС. Одна из 16 программ (SS=0000 B) обеспечивает безусловное включение всех источников, управление которыми при этом осуществля-

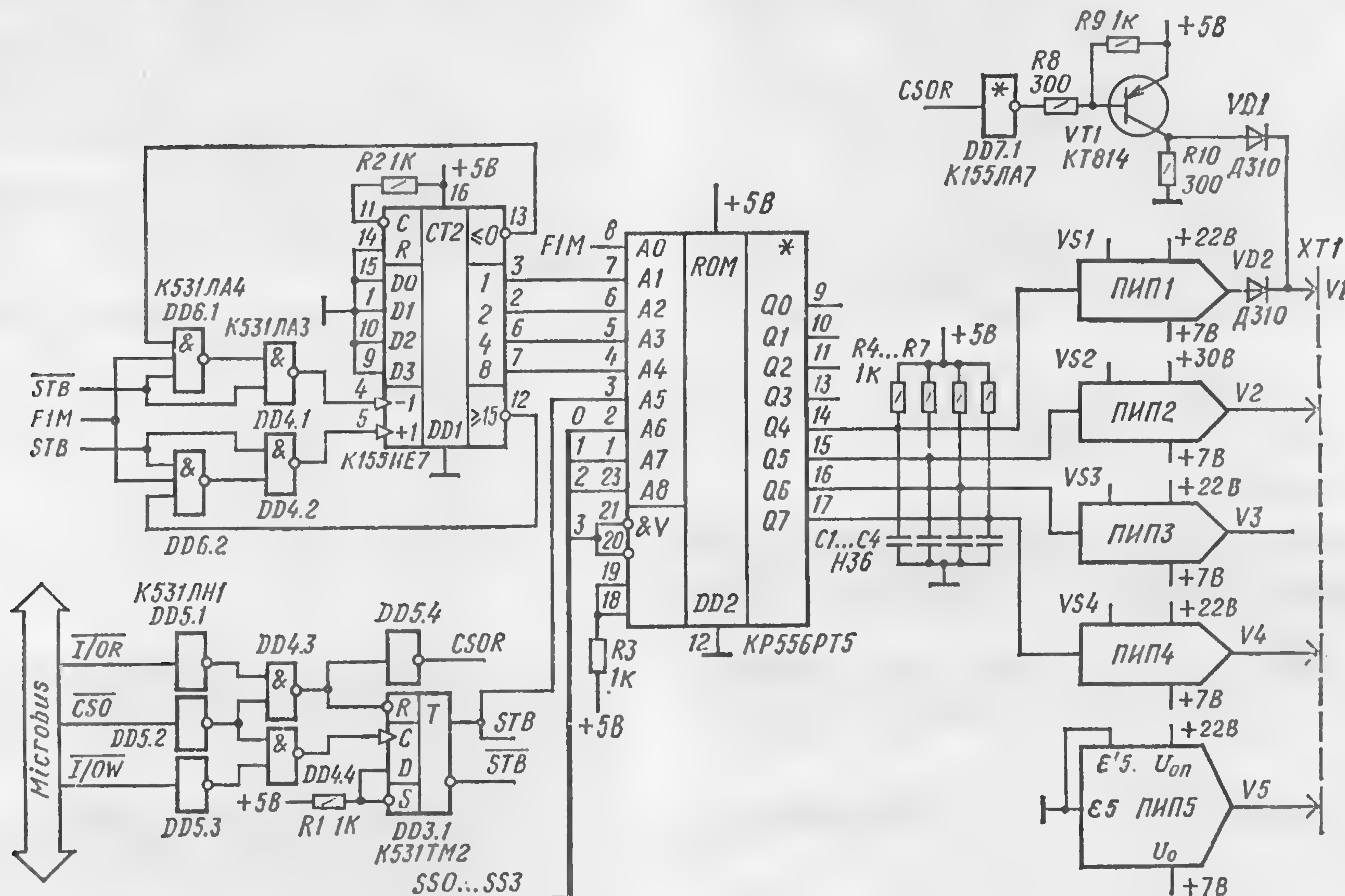


Рис. 4. Логика управления ПИП

а) F1M
STB
DD1/3
DD1/2
DD1/6
DD1/7
DD2/14 (E1)
DD2/15 (E2)
DD2/16 (E3)

б)

Адрес A8...A0, H	Данные Q7...Q0, H
062	AF
063	FF
064	8F
065	8F
...	...
07F	8F
05F	8F
05E	8F
05D	8F
05C	DF
05B	FF
Остальные	FF

Рис. 5. Работа логики управления при прожиге микросхем ПЗУ серии КР556 (SS=0001B):
а — временные диаграммы; б — таблица содержимого ППЗУ

ется чисто программным способом. Код SS выбирается во время инициализации программатора для работы с конкретной БИС.

В модуле предусмотрена возможность программного выбора скорости изменения выходного напряжения источников E1...E4. Управление осуществляется 4-разрядным полем VS, в котором для каждого источника резервируется один бит. Скорость изменения напряжения ПИП может быть снижена установкой соответствующего бита в единицу.

Аппаратная логика управления ПИП представлена на рис. 4. По спаду сигнала записи I/OW триггер DD3.1 устанавливается в единицу (STB=1). Следом за этим запускается программа включения источников, которая записана в ППЗУ DD2 и выбрана с помощью 4-разрядного поля SS. Логика рассчитана на девять программ включения-выключения. Максимальная длительность программ определяется счетчиком DD1 и равна 16 мкс. При STB=1 меандровые колебания FIM с частотой 1 МГц поступают на вход +1 счетчика. Увеличение содержимого счетчика происходит до тех пор, пока не появится сигнал переноса $\geqslant\overline{15}$. В этом состоянии счетчик остается до момента сброса триггера DD3.1.

При считывании данных (I/OR=1) триггер DD3.1 сбрасывается (STB=0) и запускает соответствующую программу выключения ПИП. Конкретный пример временных диаграмм работы логики управления для случая прожига микросхем ПЗУ серии КР556 приведен на рис. 5. Включение источников V1...V4 осуществляется сигналом «Лог. 0» на соответствующем выходе ППЗУ DD2. Для исключения влияния ложных срабатываний на выходах DD2 поставлены шунтирующие конденсаторы. Меандр FIM с частотой 1 МГц формируется счетчиком преобразователя напряжений, входящим в состав блока ПИП [2]. В каждом интервале чтения данных ключ, построенный на транзисторе VT1, формирует на выходе V1 напряжение 5В, необходимое для питания программируемой БИС. ПИП при этом выключаются.

ЛИТЕРАТУРА

1. Дианов А. П., Щелкунов Н. Н. Методика программирования микросхем ПЗУ. — Микропроцессорные средства и системы, 1985, № 3, с. 75—79.

2. Дианов А. П. Щелкунов Н. Н. Модуль программирования микросхем ПЗУ. — Микропроцессорные средства и системы, 1985, № 3, с. 80—83.

3. Лукьянов Д. А. Схемотехника универсальных программаторов ПЗУ. — Микропроцессорные средства и системы, 1985, № 3, с. 84—88.

4. Дианов А. П., Щелкунов Н. Н. Процедуры программирования логических матриц. — Микропроцессорные средства и системы, наст. номер, с. 77—80.

Статья поступила 17 июля 1985 г.

УДК 681.326.3

О. А. Щербаков

ОСОБЕННОСТИ ПРИМЕНЕНИЯ ПЛМ В МИКРОПРОЦЕССОРНЫХ СИСТЕМАХ

В предыдущих статьях «Учебного центра» показано, что для записи информации в микросхемы ПЛМ необходимо предварительно привести каждую логическую функцию F_k, задающую соответствие между выходными и входным сигналами, к логической сумме произведений входных переменных. Математически это означает, что логическая функция должна быть представлена в виде максимальной комбинации минимальных форм или, что то же самое, в дизъюнктивной нормальной форме. В большинстве случаев требуемое представление можно получить, используя формулу де-Моргана:

$$\overline{X \ast Y} = \overline{X} + \overline{Y} \quad \text{и} \quad \overline{X+Y} = \overline{X} \ast \overline{Y},$$

где знак $\ast$ означает логическое умножение, $+$ логическое сложение.

Реализация некоторых схем представлена на рис. 1. Левая колонка иллюстрирует применение обычной логики, правая — ПЛМ. Точка на пересечении линий означает наличие перемычки. В некоторых случаях приведены преобразования выходной функции по теореме де-Моргана. Существенно, что незадействованные линии термов * можно оставлять нетронутыми, как это показано на рис. 1, г. Действительно, если включить в терм как

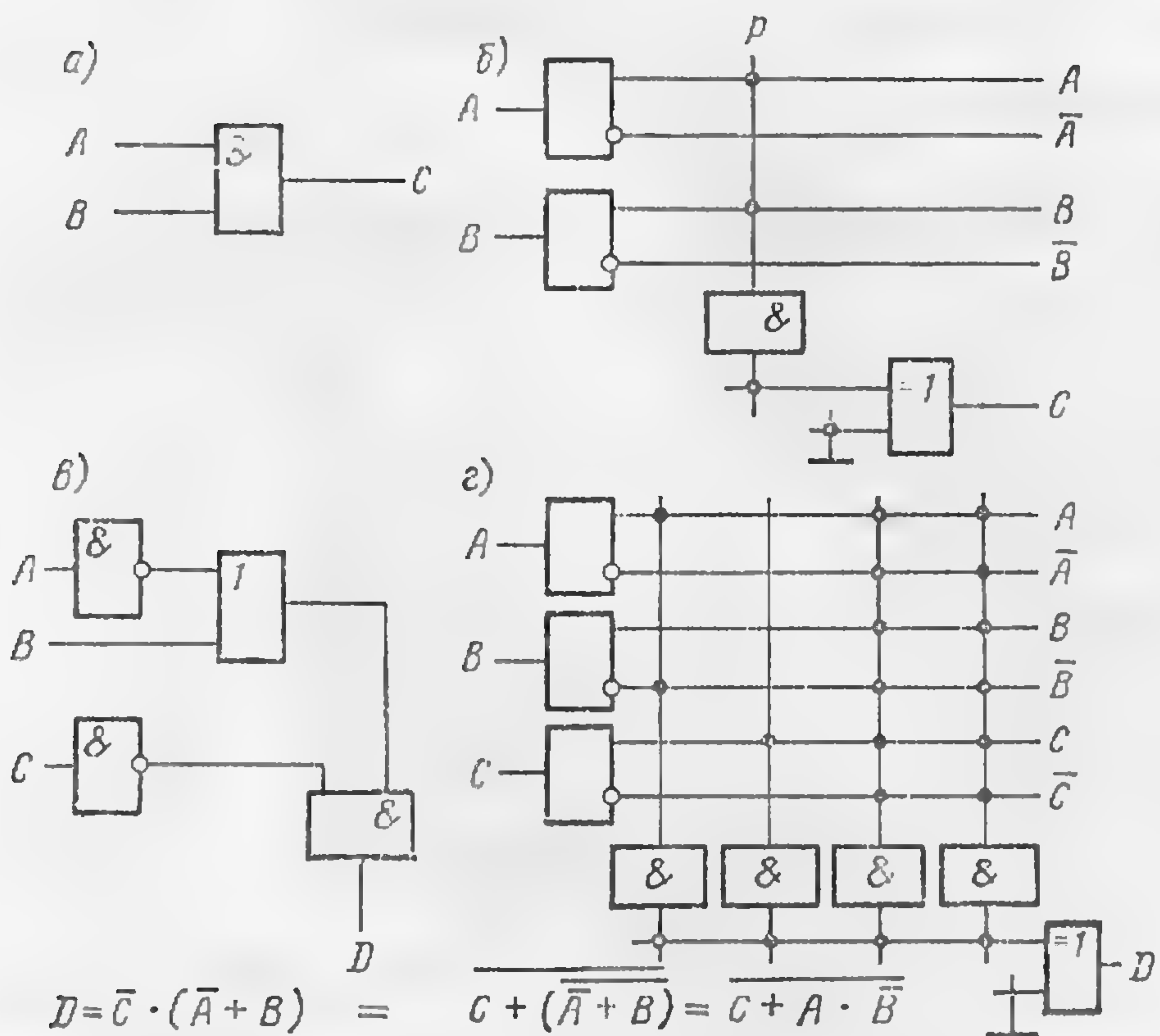


Рис. 1. Реализация электронных схем с использованием комбинаторной логики (*а*, *в*) и ПЛМ (*б*, *г*)

* Термом, или элементарной конъюнкцией, называют выражение, представляющее собой логическое произведение любого конечного множества попарно различных логических переменных, входящих в указанное выражение с отрицанием или без него.

прямое, так и инверсное значения некоторой входной переменной, то значение этого терма всегда будет равно «Лог. 0» и тем самым данный терм не будет влиять на логическую сумму. Однако неиспользуемые в активном терме входные переменные необходимо удалить путем пережигания перемычек на пересечении прямого и инверсного значений неиспользуемого входа с линией, на которой образуется данный терм.

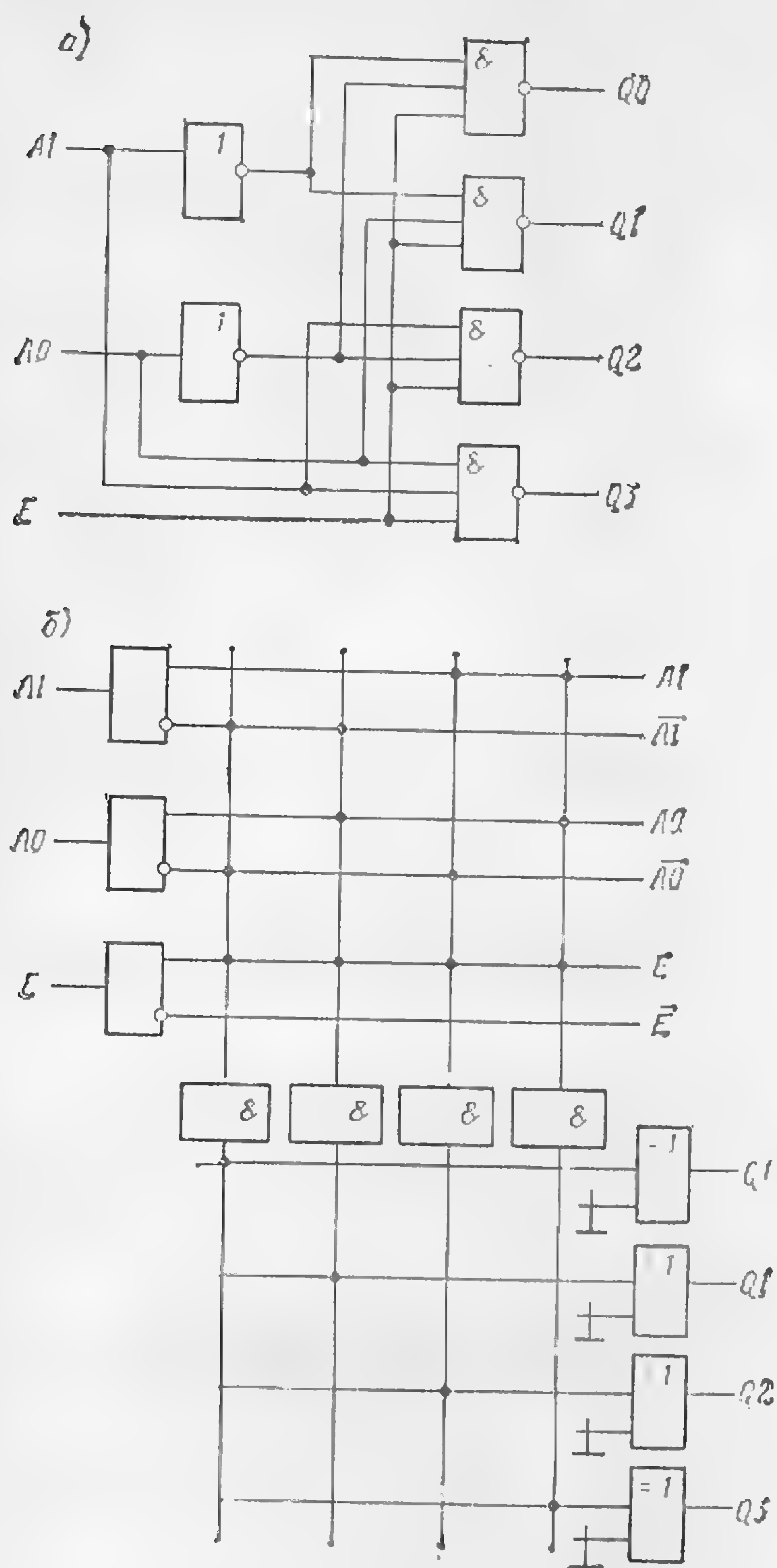
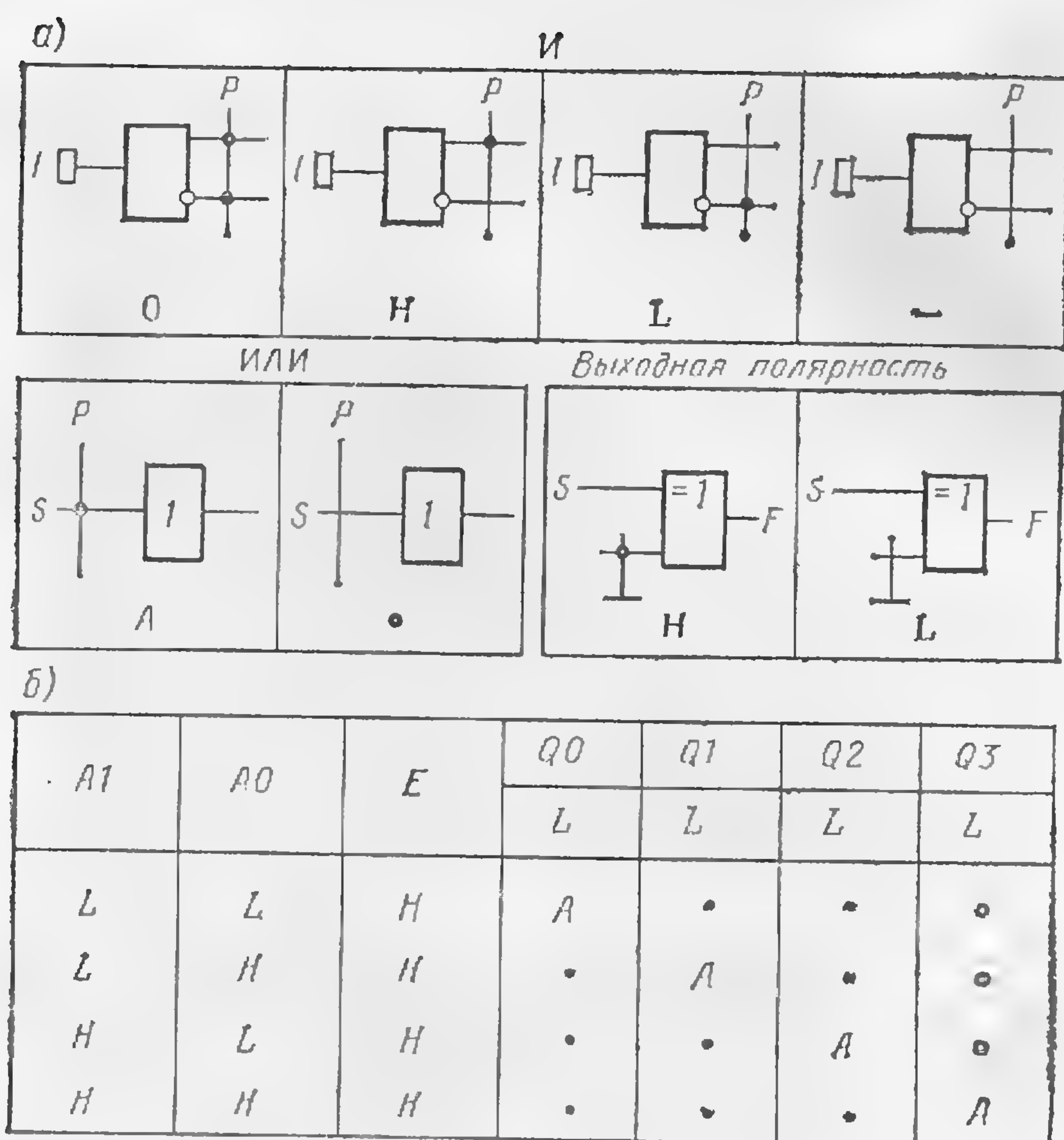


Рис. 2. Схема дешифратора на основе комбинаторной логики (*а*) и ПЛМ (*б*)

Информацию, занесенную в ПЛМ, легко редактировать и модифицировать, используя для этого все оставшиеся свободные термы. Опыт показывает, что 48 термов вполне достаточно для большинства типичных применений. При необходимости ПЛМ можно легко наращивать по входу, выходу и числу термов. Для этого служит вывод $\overline{CE}$ управления выходными буферами. Для наглядного представления логических функций при вводе их в компьютер удобно пользоваться специальной таблицей. Наличие или отсутствие определенных перемычек в ПЛМ кодируется специальными символами (рис. 3,*а*), которые затем заносятся в таблицу. Точка на пересечении линий означает присутствие плавкой перемычки. Пример заполнения таблицы для дешифратора (см. рис. 2) приведен на рис. 3,*б*.



A1	A0	E	Q0	Q1	Q2	Q3
			L	L	L	L
L	L	H	A	•	•	•
L	H	H	•	A	•	•
H	L	H	•	•	A	•
H	H	H	•	•	•	A

Рис. 3. Соответствие символов состоянию перемычек в матрице ПЛМ (*а*), таблица кодирования схемы дешифратора (*б*)

С использованием ПЛМ можно строить не только комбинационные схемы, но и схемы с памятью, способные находиться в нескольких устойчивых состояниях и в зависимости от состояния по-разному реагировать на входные сигналы. Для реализации схем подобного типа необходимо соединить некоторые входы ПЛМ с выходами и ввести тем самым обратную связь. Примером может служить Т-триггер, построенный на основе ПЛМ (рис. 4). Исключая термы, не входящие в выходные значения, и объединяя термы, обведенные пунктиром, получаем оптимизированную таблицу, приведенную в нижней правой части рис. 4. На диаграмме рядом со стрелками указаны условия перехода в другое состояние. Таким образом, аналог Т-триггера получается с использованием трех входов, двух выходов и трех термов ПЛМ. Существенно, что для исключения «гонок» в схеме Т-триггера кодирование состояний происходит так, что при переходе из одного состояния в другое меняется лишь одна двоичная цифра в его номере, т. е. кодирование ведется в коде Грэя.

Fig. 4 (г):

I			F	
C	Q*	M*	Q	M
L	L	L	•	•
H	L	L	•	A
H	L	H	•	A
L	L	H	A	A
L	H	H	A	A
H	H	H	A	•
H	H	L	A	•
L	H	L	•	•

→

I			F	
C	Q*	M*	Q	M
H	L	—	•	A
L	—	H	A	A
H	H	—	A	•

Рис. 4. Реализация Т-триггера на ПЛМ: схема соединений (*а*), временная диаграмма (*б*), диаграмма переходов (*в*), полная, оптимизированная таблица программирования ПЛМ (*г*)

Еще одним примером использования ПЛМ может служить коммутатор для выходного порта 8-разрядной микроЭВМ, от которой поступают восемь разрядов данных и строб сопровождения. В зависимости от состояния коммутатор должен выдать строб только на одно периферийное устройство и передать с этого устройства сигнал BUSY на вход микроЭВМ. В качестве периферийных устройств использовались печатающее устройство и графопостроитель. Выборка устройства, с которым в данный момент должен идти обмен, производится двумя способами: первый — переключение осуществляется с помощью кнопки; второй — задается 8-разрядным управляющим кодом (коду 11Н соответствует выборка печатающего устройства, коду 13Н — графопостроителя). Управляющий код не должен попадать в периферийное устройство, а контроллер должен при этом выдать сигнал BUSY для правильной работы схемы прерываний в микроЭВМ. Схема коммутатора приведена на рис. 5, таблица программирования — на рис. 6. Первые три терма в таблице служат для задания Т-триггера, терм 3 подавляет дребезг контактов кнопки, терм 4 запрещает выдачу строба в случае получения одного из управляющих кодов. Термы 6 и 7 осуществляют селекцию строба, 8 и 9 производят переключение, эмулируя нажатие кнопки, и, наконец, термы 10 и 11 мультиплексируют сигнал BUSY от периферийных устройств. Итак, использовав одну четверть термов ПЛМ, мы получили нужное устройство. Вариант коммутатора, построенный с использованием микросхем серии К155 и К555, требует девяти микросхем.

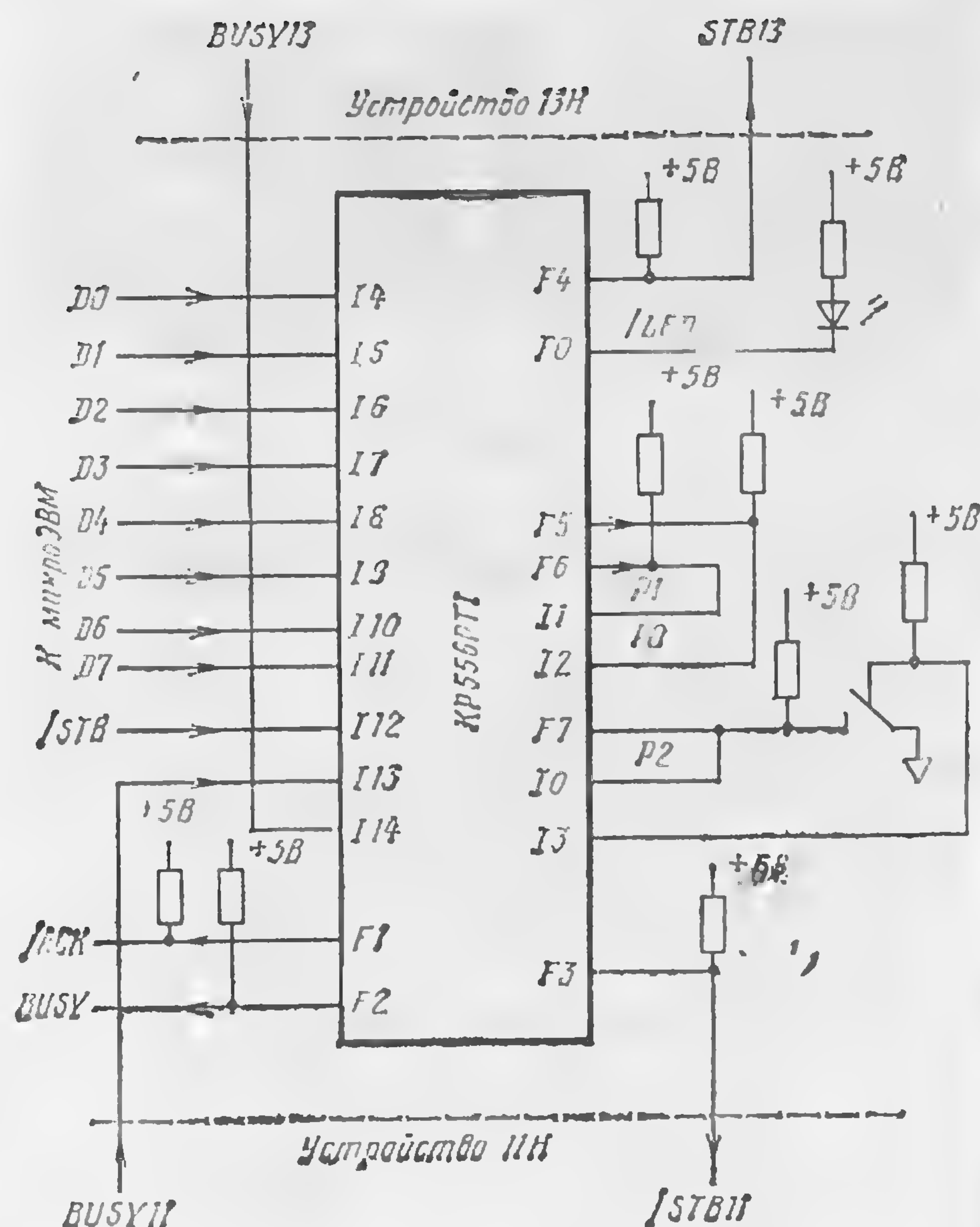


Рис. 5. Схема коммутатора на ПЛМ

N п/п	Р-терм: Входные переменные 15 (BUSY13)	14 (BUSY11)	13 (/STB)	12 (D7)	11 (D6)	10 (D5)	9 (D4)	8 (D3)	7 (D2)	6 (D1)	5 (D0)	4 (/REL)	3 (P2)	2 (P1)	1 (P0)	0	Выходная функция 7 (P2)	6 (P1)	5 (P0)	4 (/STB13)	3 (/STB11)	2 (BUSY)	1 (/ACK)	0 (/LED)
Входная полярность																	L	H	H	H	H	H	L	L
0	—	—	—	—	—	—	—	—	—	—	—	—	—	H	L	—	•	•	A	•	•	•	•	•
1	—	—	—	—	—	—	—	—	—	—	—	—	—	L	—	H	•	A	A	•	•	•	•	A
2	—	—	—	—	—	—	—	—	—	—	—	—	—	H	H	—	•	A	•	•	•	•	•	A
3	—	—	—	—	—	—	—	—	—	—	—	—	H	L	—	—	A	•	•	•	•	•	•	•
4	—	—	—	—	L	L	L	H	L	L	—	H	—	—	—	—	•	•	•	A	A	•	•	•
5	—	—	—	H	—	—	—	—	—	—	—	—	—	—	—	—	•	•	•	A	A	•	•	•
6	—	—	—	—	—	—	—	—	—	—	—	—	—	—	H	—	•	•	•	•	A	•	•	•
7	—	—	—	—	—	—	—	—	—	—	—	—	—	—	[illegible]	—	•	•	•	A	•	•	•	•
8	—	—	—	L	L	L	L	H	L	L	L	H	—	—	H	—	A	•	•	•	•	A	A	•
9	—	—	—	L	L	L	L	H	L	L	H	H	—	—	L	—	A	•	•	•	•	A	A	•
10	—	H	—	—	—	—	—	—	—	—	—	—	—	—	H	—	•	•	•	•	•	A	A	•
11	—	—	H	—	—	—	—	—	—	—	—	—	—	—	L	—	•	•	•	•	•	A	A	•
12																								
13																								
...																								

Рис. 6. Таблица программирования ПЛМ для коммутатора

ЛИТЕРАТУРА

1. Баранов С. И., Баркалов А. А. Применение программируемых логических матриц в цифровой технике. — Зарубежная радиоэлектроника, 1982, № 6, с. 67—79.

2. Integrated fuse logic, part 10, p. 366, Signetics Co., Netherlands, 1984.

Статья поступила после переработки 10 ноября 1985 г.

Микросхема статического ОЗУ КМ132РУ5

Интегральная микросхема КМ132РУ5 представляет собой оперативное запоминающее устройство статического типа с произвольной выборкой, выполненное по n-канальной МДП-технологии. Информационная емкость ОЗУ 4К бит. ОЗУ совместимо по входным и выходным сигналам со схемами ТТЛ. Для питания микросхемы необходим один источник с напряжением 5 В±10 %. Кристалл ОЗУ заключен в 18-выводной металлокерамический корпус (рис. 1) с размерами 22,2×7,5 мм. Назначение выводов показано в табл. 1.

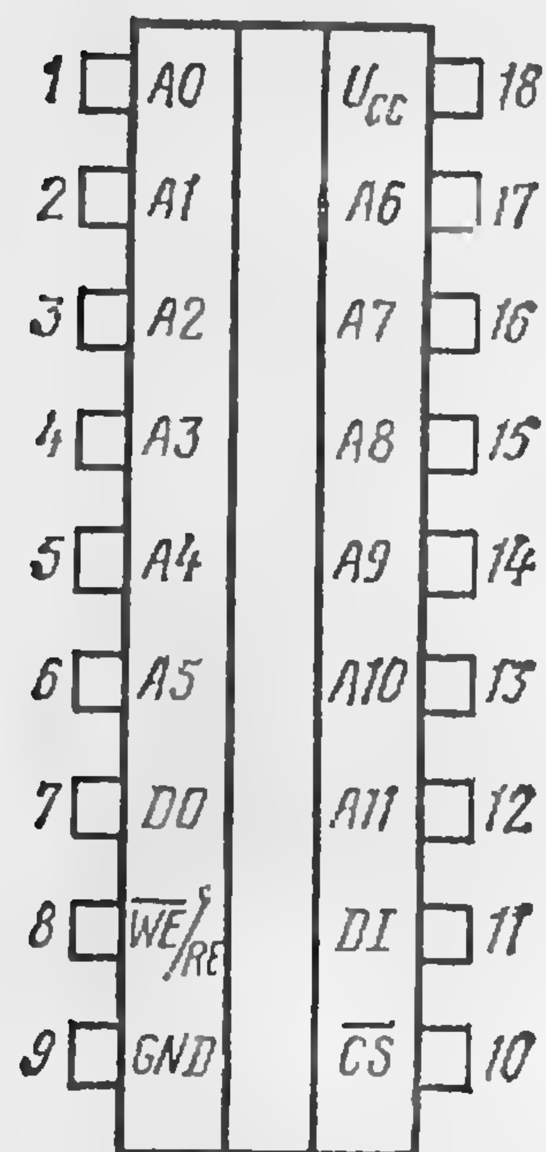


Рис. 1. Условное графическое изображение микросхемы КМ132РУ5

Таблица 1

Назначение выводов

Вывод	Назначение
1 ... 6	Адресные входы столбца A0 ... A5
7	Информационный выход DO
8	Вход сигнала «запись — считывание» $\overline{WE}$/RE
9	Общий вывод GND
10	Вход сигнала «выбор микросхемы» $\overline{CS}$
11	Информационный вход DI
12 ... 14	Адресные входы столбца A11 ... A9
15 ... 17	Адресные входы строки A8 ... A6
18	Напряжение источника питания U_{cc}+; +5 В

Для увеличения быстродействия и уменьшения влияния подложки на кристалле имеется встроенный преобразователь напряжения отрицательного смещения на подложку ($U_{см}$=−2,7 В).

Микросхема содержит усилители адреса по строкам и столбцам, усилители сигналов «входная информация», «запись — считывание», «выбор микросхемы», дешифраторы строк и столбцов, накопитель на 4096 запоминающих ячеек, 64 усилителя ввода-вывода, усилитель выходных данных, преобразователь напряжения отрицательного смещения на подложку (рис. 2).

Запоминающая ячейка памяти представляет собой 6-транзисторный статический триггер. Накопитель ОЗУ состоит из двух блоков ЗУ емкостью 2048 бит и одного дешифратора строк. Микросхема с полной дешифровкой требует 12 адресных входных шин (A0...A11) для выбора одного из 4096 хранимых слов. Все входные сигнальные шины имеют буферные входные усилители. Раздельные шины входной и выходной информации устраняют взаимное влияние циклов записи и считывания. Выход микросхемы имеет три рабочих состояния в зависимости от состояния ключей, определяемого присутствием команд $\overline{CS}$ и $\overline{WE}$/RE.

Выходная информация из накопителя для хранимой «Лог. 1» соответствует выходному напряжению «Лог. 1», а для хранимого «Лог. 0» — выходному напряжению «Лог. 0» на выходе DO микросхемы. Максимальный входной уровень «Лог. 0» по всем входам 0,8 В, минимальный входной уровень «Лог. 1» 2 В. Временные диаграммы работы микросхемы в режимах считывания и записи приведены на рис. 3...6, временные параметры сигналов — в табл. 2.

В режиме считывания на вход DI микросхемы может подаваться любой код. Отсчет всех временных параметров проводится по уровню 1,5 В. Напряжение «Лог. 0» и «Лог. 1» сигналов A0...A11, $\overline{CS}$, $\overline{WE}$/RE, DI должны быть в пределах: U_L=0...0,4 В; U_H=2,4... ...5,5 В. Для обеспечения максимального быстродействия микросхемы (t_{AA}, t_{ACS1}, t_{ACS2}, t_{RC1}, t_{RC2}) длительность фронтов (t_{HL}, t_{LH}) сигналов A0...A11, $\overline{CS}$ должна быть

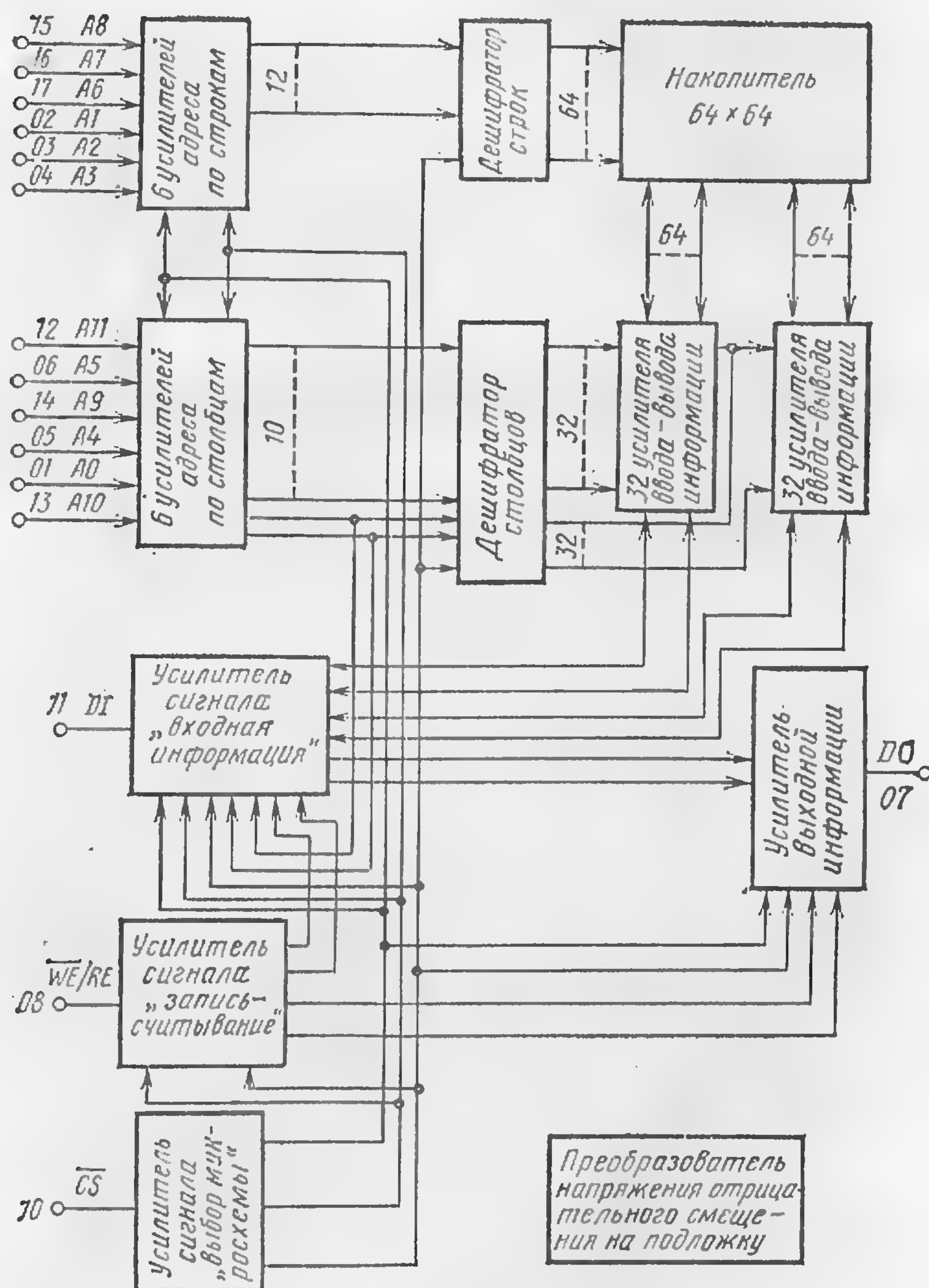


Рис. 2. Электрическая структурная схема

Таблица 2

Временные параметры сигналов микросхемы КМ132РУ5, Т = —10...+70 °С

Наименование	Обозначение	Норма, нс, для микросхемы	
		КМ132РУ5А	КМ132РУ5Б
Время цикла записи	t_{WC}	85	120
Время цикла считывания ($U_{CSL} \leqslant 0{,}4$ В или $t_{CS} \geqslant 55$ нс, $C_н = 30$ пФ)	t_{RC1}	85	120
Время цикла считывания ($0_{нс} < t_{CS} < 55$ нс, $C_н = 30$ пФ)	t_{RC2}	100	130
Время выборки адреса ($U_{CSL} < 0{,}4$ В, $C_н = 30$ пФ)	t_{AA}	85	120
Время выборки сигнала выбора микросхемы ($t_{CS} \geqslant 55$ нс, $C_н = 30$ пФ)	t_{ACS1}	85	120
Время выборки сигнала выбора микросхемы ($0_{нс} < t_{CS} < 55$ нс, $C_н = 30$ пФ)	t_{ACS2}	100	130
Время удержания высокого импеданса относительно сигнала выбора микросхемы	t_{LZ}	10	10
Время сохранения выходной информации после изменения сигнала адреса	t_{OH}	5	5
Время сохранения выходной информации после сигнала выбора микросхемы	t_{HZ}	—	0...40
Длительность фронтов	t_{HL}, t_{LH}	0...40	—
Длительность сигнала адреса	t_A	85	120
Длительность сигнала выбора микросхемы	$t_{\overline{CS}}$	75	110
Время установления сигнала выбора микросхемы относительно сигнала адреса	t_{AS}	0	0
Время восстановления сигнала выбора микросхемы относительно сигнала адреса	t_{CSR}	15	15
Время удержания сигнала записи относительно сигнала адреса	t_{AW}	75	110
Время установления сигнала записи относительно сигнала адреса	t_{WSA}	0	0
Длительность сигнала записи	t_{WP}	55	60
Время восстановления сигнала записи относительно сигнала адреса	t_{WR}	15	15
Время удержания сигнала записи относительно сигнала входной информации	t_{DW}	35	50
Время сохранения сигнала входной информации после сигнала записи	t_{DH}	10	10
Время сохранения сигнала выходной информации после сигнала считывания	t_{WZ}	0...35	0...50
Время сохранения высокого выходного импеданса после сигнала записи	t_{OW}	0	0

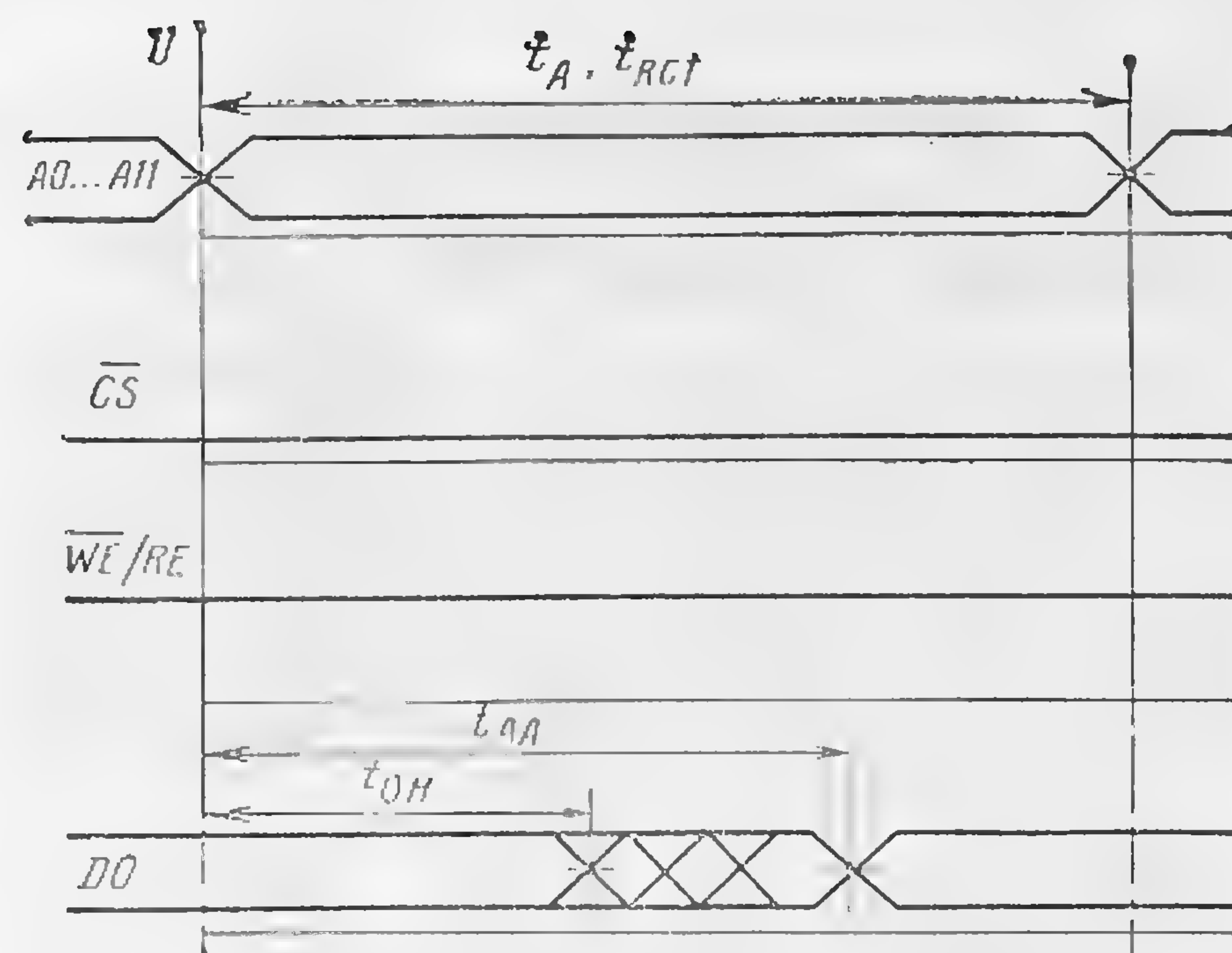


Рис. 3. Режим 1 считывания

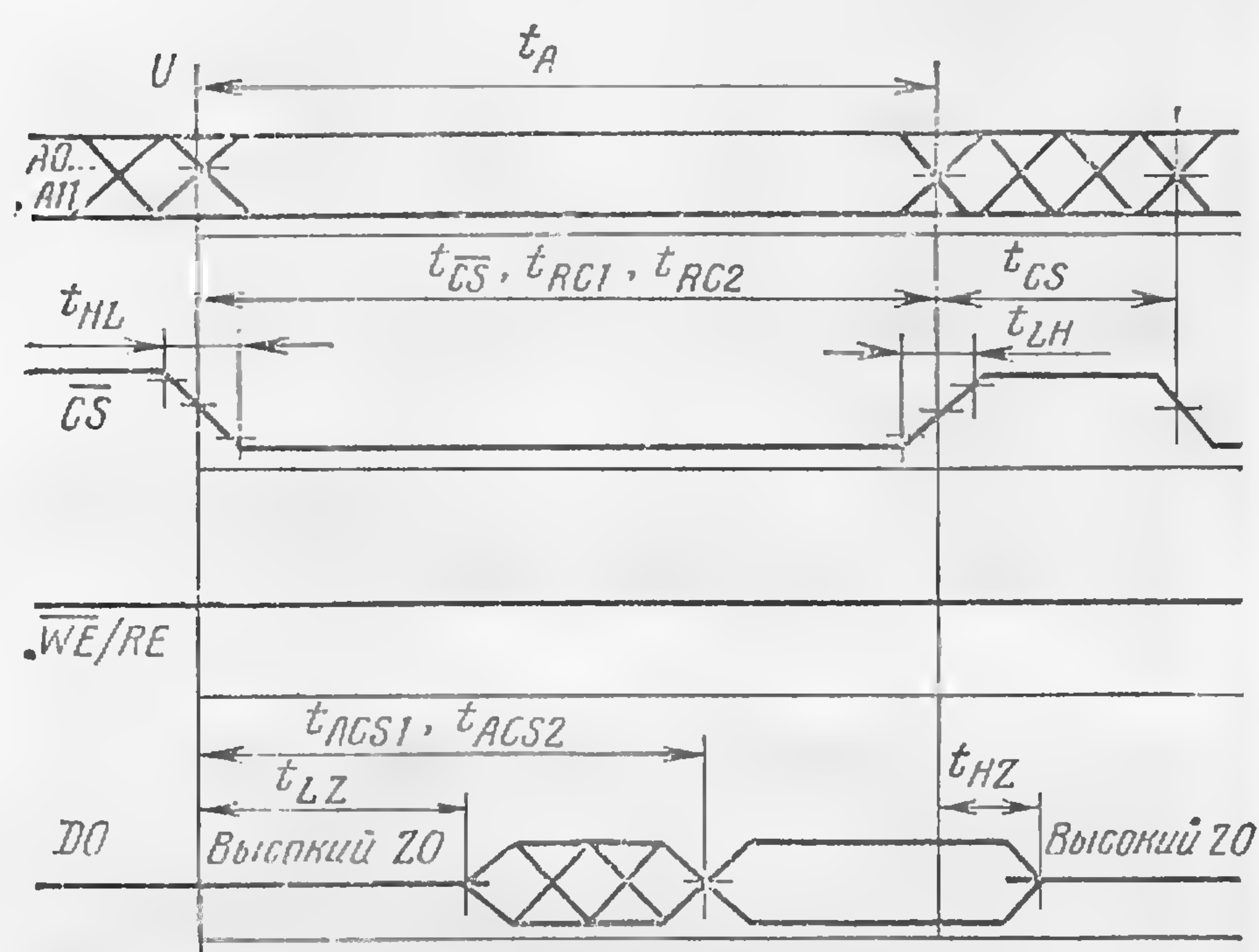


Рис. 4. Режим 2 считывания

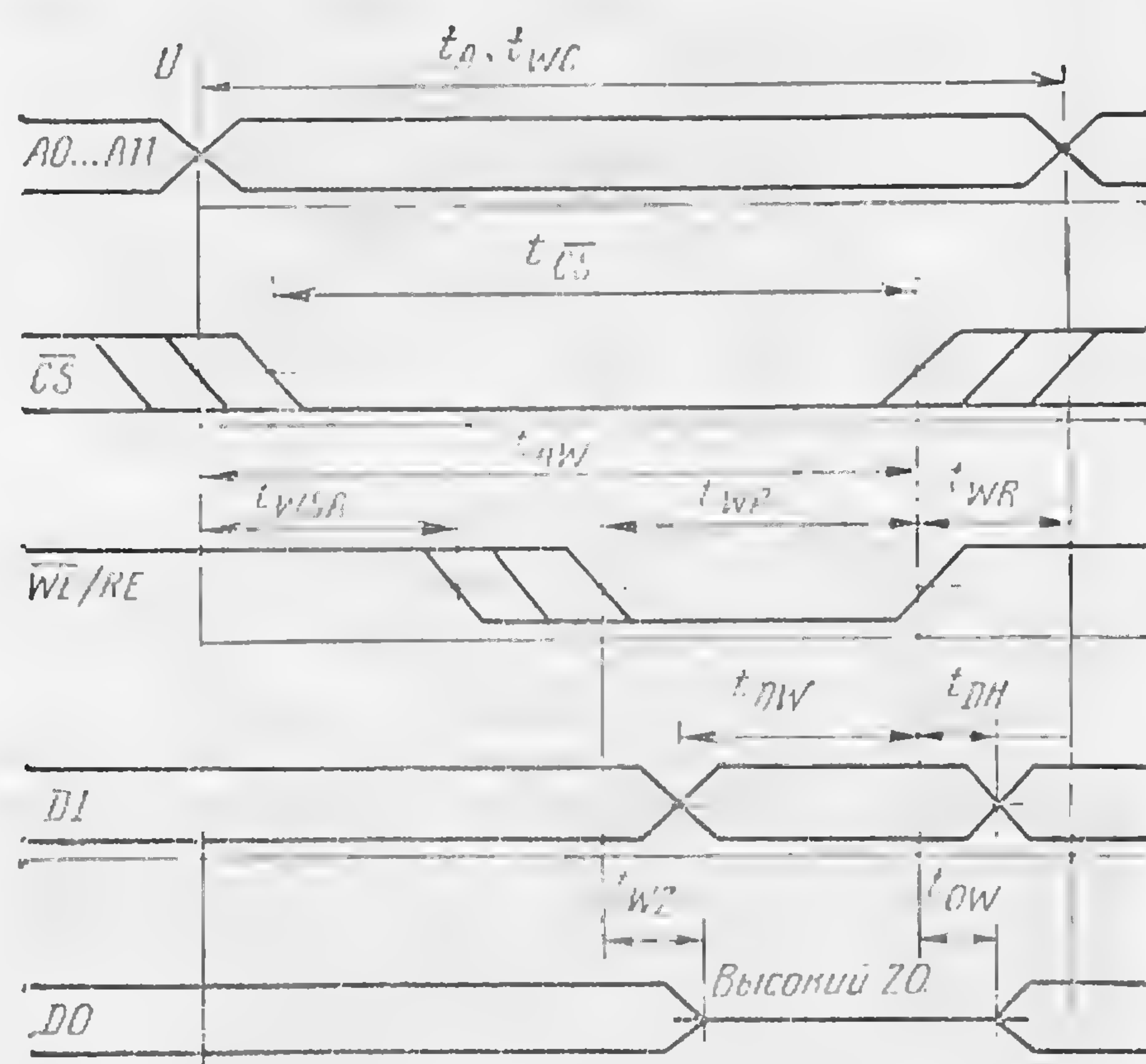


Рис. 5. Режим 1 записи

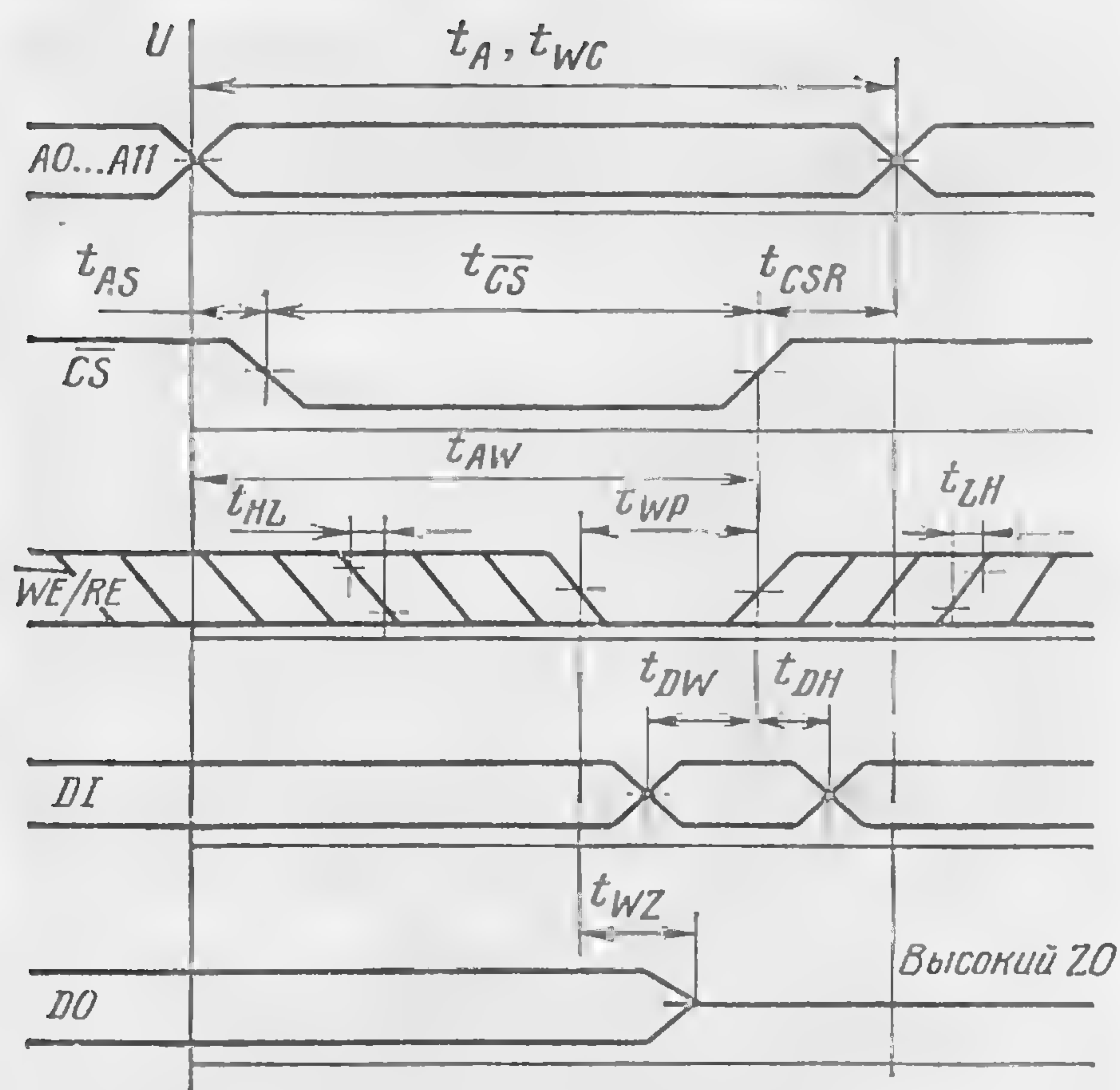


Рис. 6. Режим 2 записи

не более 10 нс. Основные электрические параметры микросхемы представлены в табл. 3, предельные значения эксплуатационных параметров показаны в табл. 4.

При эксплуатации микросхем должны быть приняты меры, исключающие возможность накопления электростатических зарядов на выводах. Допустимая величина статического потенциала не более 100 В. Коэффициент объединения по выходу не более 40 с нагрузкой из трех входов ТТЛ-схем, например, серии К155. Неиспользуемые входы подключаются к источнику питания U_{cc} или к общему выводу GND.

Формовка выводов не допускается. Для каждой микросхемы необходимо в непосредственной близости от выводов 18 и 9 подключить керамический конденсатор емкостью не менее 0,1 мкФ, а для группы (16 микросхем) — дополнительно электролитический конденсатор емкостью не менее 20 мкФ.

Гарантийная наработка 15 000 ч в пределах гарантийного срока хранения (15 лет).

Таблица 3

Основные электрические параметры микросхем КМ132РУ5
T = −10 ... +70 °C

Наименование параметра, единица измерения	Обозначение	Норма для микросхемы*
Напряжение низкого уровня сигнала выходной информации, В, не более ($I_{DOL} \leqslant 5$ А)	U_{DOL}	0,45
Напряжение высокого уровня сигнала выходной информации, В, не менее ($I_{DOH} \leqslant$ $\leqslant 1$ А)	U_{DOH}	2,4
Ток потребления динамический, мА, не более ($U_{CSL} \leqslant$ $\leqslant 0,4$ В)	I_{CCH}	180

* Данные справедливы для групп А и Б.

Таблица 4

Предельные значения эксплуатационных параметров

Наименование параметра, единица измерения	Обозначение	Норма	
		мин.	макс.
Напряжение питания относительно общего вывода, В	U_{CC}	—	6
Напряжение на любом выводе микросхемы относительно общего вывода, В	U	−1	6
Выходной ток, мА	I_{DOL}, I_{DOH}	—	20
Емкость нагрузки, пФ	C_H	—	600

Рекомендации по пайке паяльником

Температура жала паяльника, °С, не более . . . 280
Время касания каждого вывода, с, не более . . . 3
Интервал между пайками соседних выводов, с, не менее 10
Расстояние от корпуса до места пайки (по длине вывода), мм, не менее 1

Рекомендации по групповой пайке

Температура припоя, °С, не более 265
Время воздействия припоя одновременно на все выводы, с, не более 2
Интервал между двумя повторными пайками одной микросхемы, мин, не менее 5
Расстояние от корпуса до места пайки, мм, не менее 1

Окончание. Начало см. на стр. 21

Дополнительным узлом фильтров является сдвиговый регистр комплексного числа. В первом случае $P\dot{X}$ должен задерживать на столько тактов, какова разрядность числа $\dot{X}$. $P\dot{Y}$ имеет длину во столько же тактов, что и $P\dot{X}$, за вычетом числа тактов задержки результата $\dot{Z}$ по отношению к отсчету $\dot{X}_n$. Для того чтобы комплексные числа на входы $\dot{X}$, $\dot{Y}$ НРФ приходили одновременно, отсчет $\dot{X}_n$ снимается с $P\dot{X}$ задержанным на то же время, что и результат на выходе $\dot{W}$ по отношению к операндам на входах $\dot{U}$, $\dot{V}$.

На рис. 5 показана схема устройства восьмиточечного быстрого преобразования Фурье на двенадцати процессорах БПФ. Для его реализации требуется 48 корпусов МП К1815ВФ3. Рассеиваемая устройством мощность составляет 17 Вт, а производительность для 20-разрядных чисел достигает 6 млн. базовых операций БПФ в секунду. Применяя для построения цифровых фильтров соответствующие методы распараллеливания и каскадные схемы реализации фильтров высокого порядка, можно эффективно осуществлять разнообразную обработку сигналов.

Микропроцессор К1815ВФ3 позволяет создавать устройства, имеющие в 10...100 раз более высокую производительность при меньших габаритах и более низкой потребляемой мощности по сравнению с устройствами, например, на основе МПК БИС серии К1802.

ЛИТЕРАТУРА

1. Каляев А. В. Микропроцессорные вычислительные системы с программируемой архитектурой. — В кн.: Многопроцессорные вычислительные структуры. — Таганрог, 1981, вып. 3 (XII), с. 5—12.
2. Рабинер Л., Гоулд Б. Теория и применение цифровой обработки сигналов. — М.: Мир, 1978. — 848 с.

Статья поступила 30 августа 1985 г.

ПРОГРАММА КУРСА «ОСНОВЫ ИНФОРМАТИКИ И ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ»

X—XI классы (102 ч)

Публикуемая программа курса «Основы информатики и вычислительной техники» предназначена для конкурса на создание учебника для учащихся общеобразовательной школы. Это определяет ее особенности по сравнению с обычными учебными программами: в содержании методические предложения сведены до минимума, чтобы авторы учебника были свободны в использовании методических приемов изложения курса.

Программа курса разрабатывалась исходя из понимания основ информатики и вычислительной техники **как общеобразовательного учебного предмета.** Его содержание базируется на определении информатики как научной дисциплины, имеющей своим предметом изучение законов и методов накопления, передачи и обработки информации с помощью ЭВМ.

Программа предполагает, что в ближайшем будущем школы страны будут оснащены вычислительной техникой, предоставляющей учащимся новое средство познания и моделирования реального мира, работа с которым должна способствовать их всестороннему развитию, дать им знания и умения, необходимые как в период обучения, так и после окончания школы в трудовой деятельности и при продолжении образования.

Вышесказанное определяет требования к курсу «Основы информатики и вычислительной техники». Этот курс должен формировать у учащихся:

— навыки грамотной постановки задач, возникающих в практической деятельности, для их решения с помощью ЭВМ;

— навыки формализованного описания поставленных задач, элементарные знания о методах математического моделирования и умение строить простые математические модели поставленных задач;

— знания основных алгоритмических структур и умение применять эти знания для построения алгоритмов решения задач по их математическим моделям;

— понимание устройства и функционирования ЭВМ и элементарные навыки составления программ для ЭВМ по построенному алгоритму на одном из языков программирования высокого уровня;

— навыки квалифицированного использования основных типов современных информационных систем для решения с их помощью практических задач и понимания основных принципов, лежащих в основе функционирования этих систем;

— умение грамотно интерпретировать результаты решения практических задач с помощью ЭВМ и применять эти результаты в практической деятельности.

Эти требования, взятые в их минимальном объеме, составляют задачу достижения первого уровня компьютерной грамотности, взятые в максимальном объеме — составляют перспективную задачу — воспитание информационной культуры учащихся.

Представляется, что учебник по курсу «Основы информатики и вычислительной техники», содержание которого определяется данной программой, должен решать задачу ближайшего будущего, задачу достижения первого уровня компьютерной грамотности, как она сформулирована в директивных документах партии и правительства.

Программа курса рассчитана на обучение основам информатики в IX—X (X—XI) классах в объеме 102 ч. При этом максимальный объем рукописи учебника не должен превышать 18 п. л.

Номенклатура и последовательность расположения тем в программе курса обязательны для авторов учебника. Вместе с тем, содержание каждой темы может быть изменено авторами, в зависимости от принятой ими методики раскрытия материала.

Программа предусматривает циклический принцип развития курса. Темы 1—3 задают начальный уровень понятий, а в темах 4—7 происходит их дальнейшее развитие и закрепление. Такая структура курса определяется тесной взаимосвязью основных понятий информатики, не позволяющей предложить учащимся сколько-нибудь подробный материал по одному из них без наличия у школьников хотя бы минимальных знаний по остальным. Кроме того, циклическое развитие курса соответствует педагогическим традициям советской школы и оправдало себя в практике обучения другим школьным учебным предметам.

Выделение в программе специального языка для записи алгоритмов преследует цель обеспечения всех учащихся равной подготовкой в вопросах построения алгоритмов. Алгоритмический язык позволяет выделить структуру алгоритма, осуществить систематическую разработку алгоритма, отработать методы алгоритмизации, «не привязываясь» к конкретной ЭВМ. Вместе с тем он позволяет осуществить достаточно легкий переход к построению программы на конкретном языке программирования.

Выбор конкретного языка (языков) программирования, работе с которым посвящена пятая тема данной программы, предоставлен авторам учебного пособия.

Перечни содержаний и видов практических работ, программного обеспечения и состав межпредметных связей носят рекомендательный характер.

Приведенный в программе перечень знаний и умений определяет желаемый уровень подготовки учащихся, к которому должны стремиться авторы учебника.

В разделе «Требования к знаниям и умениям» был сделан естественный упор на умения, что должно служить повышению активности учащихся на уроках, приобретению ими навыков работы с вычислительной техникой.

Межпредметные связи курса основ информатики и вычислительной техники имеют в нем важное методологическое значение. Проработка этих связей в настоящей программе сделана в самом общем виде для того, чтобы, с одной стороны, подчеркнуть важность этого элемента курса, а с другой — предложить авторам учебника достаточно широкую свободу в выборе форм и методов раскрытия межпредметных связей школьного курса информатики.

Важнейшая особенность нового курса основ информатики, методики и организации учебного процесса при его изучении — постоянная работа школьников с ЭВМ на каждом уроке информатики.

Все темы курса включают в себя значительный объем практических работ с использованием ЭВМ, при котором «контактное время» работы с ЭВМ составляет не менее половины урока. В курсе предусматриваются три вида организованного использования кабинета вычислительной техники на уроках информатики — демонстрация, лабораторная работа и практикум.

Демонстрация. Работу у ЭВМ ведет учитель; учащиеся либо наблюдают за его действиями через демонстрационный экран, либо воспроизводят эти действия на своих рабочих местах.

Лабораторная работа (фронтальная). Сравнительно короткий период самостоятельной, но синхронной работы учащихся с учебным программным средством, направленной либо на его освоение, либо на закрепление материала, объясненного учителем, либо на проверку усвоения полученного знания или операционного навы-

ка. Роль учителя во время фронтальной лабораторной работы — обеспечение синхронности действий учащихся и оказание экстренной помощи по инициативе учеников.

Практикум. Выполнение протяженной самостоятельной работы с компьютером в пределах одного-двух уроков по индивидуальному заданию; работа требует синтеза знаний и умений по целому разделу курса; элементы синхронизации работы во время практикума сведены к минимуму. Учитель, главным образом, обеспечивает индивидуальный контроль за работой учащихся.

Учитывая организацию учебной работы в кабинете вычислительной техники, когда все школьники имеют перед собой персональные дисплеи с экраном, можно предположить, что демонстрация потеряет свой традиционный смысл. Она, как правило, станет разновидностью фронтальной практической работы. Поэтому, в проекте программы демонстрации объединены с фронтальными лабораторными работами в единую рубрику «Фронтальные практические работы».

Постоянное использование компьютера в процессе обучения информатике требует поддержки будущего учебника развитым программным обеспечением.

При создании учебных программных средств особое внимание должно быть уделено интеграции их в единый комплекс, в частности, программное обеспечение должно иметь единый унифицированный интерфейс. В рамках одного проекта курса видится обоснованным требование совместимости программного обеспечения по языковым системам программирования; должна быть предусмотрена возможность взаимодействия различных учебных пакетов.

Авторским коллективам следует иметь в виду, что учебник по курсу информатики должен выполнять функцию организации учебной деятельности школьников в условиях широкого использования ЭВМ на уроках. Наряду с изложением теоретического материала, задачами и упражнениями, учебник должен содержать все описания и инструкции, необходимые для работы с программным обеспечением ЭВМ. Это положение не исключает разработки для учащихся (и учителей) дополнительной литературы по использованию базового программного обеспечения школьной ЭВМ.

1. Введение

Что изучает информатика? Понятие об информации (знания и данные) и ее обработке. Вычислительная техника как средство автоматической обработки информации.

Примеры работы ЭВМ.

Роль информатики и вычислительной техники на современном этапе развития общества.

Фронтальные практические работы: диалог с ЭВМ; игра с ЭВМ.

Межпредметные связи. При изучении этой темы следует черпать примеры представления, хранения и обработки информации, алгоритмического решения задач из всего школьного опыта: пользование словарями на уроках языка, указателями и таблицами на уроках истории и географии, кодирование информации азбукой Морзе, передача информации по телефону в физике, алгоритмы арифметических действий в математике, рецептурные правила рабочих процедур в природоведении, географии, химии, физике, примеры вычислений с помощью микрокалькулятора в алгебре.

Требования к знаниям и умениям. Учащиеся должны знать:

— определение предмета информатики.

Учащиеся должны уметь приводить примеры:

— накопления, хранения, передачи и обработки информации в профессиональной деятельности человека и повседневной жизни;

— обработки информации в естественных системах;

— применения ЭВМ.

2. Первоначальное знакомство с ЭВМ

Общий вид ЭВМ, основные устройства компьютера, их функции и взаимосвязь в процессе работы машины. Представление о программном управлении работой ЭВМ. Взаимодействие человека и ЭВМ.

Школьная ЭВМ, локальная сеть в кабинете вычислительной техники. Правила техники безопасности при работе на ЭВМ. Знакомство с клавиатурой микроЭВМ.

Первоначальный опыт работы на ЭВМ. Компьютер как: вычислительное устройство, инструмент моделирования, средство хранения и систематизации информации, управляющее устройство. Представление о программном обеспечении ЭВМ.

Фронтальные практические работы: начальные навыки работы на клавиатуре ЭВМ; ввод и редактирование текста; построение графического изображения; использование «меню»; решение параметрической задачи по готовой программе; взаимодействие с ЭВМ в масштабе реального времени.

Межпредметные связи. При объяснении устройства ЭВМ следует опираться на сведения из курса физики. Демонстрационные примеры и содержательные задачи для практических работ могут быть взяты из подходящих школьных предметов: вычислительная задача из курсов физики или химии, работа с текстом из курсов языка и литературы, построение изображения из курса геометрии или черчения, развивающие игры по разным предметам в качестве примера взаимодействия с ЭВМ.

Требования к знаниям и умениям. Учащиеся должны знать:

— название и назначение основных устройств ЭВМ;

— правила техники безопасности при работе на ЭВМ.

Учащиеся должны уметь:

— начинать и завершать работу на ЭВМ;

— пользоваться клавиатурой и курсором;

— исполнить в режиме диалога простую программу на ЭВМ, применяя «меню», запросы о «помощи» и инструкции к пользованию;

— набрать простой текст в пределах емкости экрана;

— построить простейшее изображение с помощью графического редактора.

3. Основы алгоритмизации

Понятие об алгоритме. Исполнитель алгоритма. Система команд исполнителя. Примеры алгоритмов и исполнителей. Представление о свойствах алгоритмов. Возможность автоматизации исполнения алгоритмов.

Алгоритмический язык как средство представления и записи алгоритмов. Правила записи алгоритмов на алгоритмическом языке. Последовательное выполнение команд. Вспомогательные алгоритмы. Условия в алгоритмах. Команды ветвления и повторения.

Алгоритмы работы с величинами. Понятие величины (имя и значение). Постоянные и переменные величины. Типы величин: числовые, текстовые, табличные. Действия над величинами. Аргументы и результаты алгоритма. Команда присваивания. Промежуточные величины. Отношения между величинами в качестве условий. Алгоритмы вычисления функций. Исполнение алгоритмов. Таблица значений.

Построение алгоритмов. Условие задачи и ее математическая модель: формулы, неравенства, рекуррентные соотношения. Метод последовательных уточнений. Рекурсивный алгоритм. Библиотека алгоритмов и ее использование.

Фронтальные практические работы: правила использования учебного программного обеспечения; построение и исполнение алгоритма для заданного исполнителя; запись и выполнение основных конструкций алгоритмического языка; построение и исполнение алгорит-

ма работы с величинами; построение и исполнение алгоритма работы с таблицами; построение и исполнение алгоритма вычисления функции; построение и исполнение рекурсивного алгоритма; построение и исполнение алгоритма с использованием вспомогательных алгоритмов.

Практикум. 1. Составление и исполнение алгоритмов с использованием заданного исполнителя. 2. Составление и исполнение алгоритмов с использованием основных команд алгоритмического языка. 3. Итоговый практикум по алгоритмическому языку.

Межпредметные связи. Основу межпредметных связей по данной теме составляет номенклатура задач, для решения которых строятся алгоритмы и программы. Понятие величины в алгоритмах вводится на основе и в сравнении с величинами в физике и математике. Алгоритмы вычисления функций могут расширить представление о понятии математической функции. В качестве примеров математических моделей можно брать формулы и соотношения из алгебры, тригонометрии, выражения физических закономерностей.

Требования к знаниям и умениям. Учащиеся должны знать:

— содержание понятия алгоритма и его основных свойств;

— название, правила записи основных конструкций алгоритмического языка; основные типы величин; правила выполнения команд и исполнения алгоритмов.

Учащиеся должны уметь:

— обнаруживать типичные формальные ошибки в записи алгоритмов;

— записывать на алгоритмическом языке несложные алгоритмы, аналогичные описанным в учебнике;

— проследить исполнение алгоритма с использованием таблицы значений;

— применить метод последовательных уточнений при построении алгоритмов, в том, числе, с использованием библиотеки; построить простой рекурсивный алгоритм по заданному рекуррентному соотношению.

4. Основы вычислительной техники

Архитектура ЭВМ. Функциональная организация ЭВМ: процессор, память, внешние устройства, команды, данные.

Представление информации в ЭВМ. Принцип двоичного кодирования информации. Представление данных. Понятие о системах счисления. Представление команд.

Процессор. Арифметико-логическое устройство, устройство управления, счетчик команд. Оперативная память. Магистраль. Организация автоматического исполнения программы. Типы машинных команд.

Представление о логических элементах ЭВМ.

Внешние устройства ЭВМ. Внешняя память, клавиатура, дисплей, принтер.

Типы ЭВМ: программируемый микрокалькулятор, микроЭВМ, мини-ЭВМ, универсальные ЭВМ, супер-ЭВМ. Сети ЭВМ.

Фронтальные практические работы. Работа с программными моделями ЭВМ: двоичное кодирование информации; исполнение машинных команд; работа логических элементов ЭВМ; работа внешних устройств ЭВМ.

Межпредметные связи. Сведения о физических принципах устройства ЭВМ опираются на различные разделы курса физики. Представление о двоичной системе счисления, работе арифметического устройства связывается с соответствующими разделами математики.

Требования к знаниям и умениям. Учащиеся должны знать:

— функциональную организацию ЭВМ; принцип автоматического исполнения программы;

— принципы представления данных и команд в ЭВМ; особенности двоичной системы счисления;

— названия и примерные характеристики основных типов ЭВМ.

Учащиеся должны уметь:

— объяснять функциональную организацию ЭВМ и принципы действия устройств машины (по плакату);

— приводить примеры машинных команд.

5. Основы программирования

ЭВМ как универсальный исполнитель с развиваемой системой команд. Программное обеспечение ЭВМ.

Базовое программное обеспечение. Операционная система. Система программирования. Файловая система. Графическая система.

Язык программирования. Данные, величины, команды. Реализация алгоритмических конструкций в языке программирования.

Организация исполнения программы на ЭВМ.

Понятие об интерпретаторах и компиляторах.

Фронтальные практические работы: с операционной, файловой и графической системами; обращение к ЭВМ за помощью; реализация конструкций алгоритмического языка в языке программирования; вызов библиотечных подпрограмм; ввод и вывод данных; построение и отладка программ.

Практикум. 1. Работа с базовым программным обеспечением. 2. Итоговый практикум по программированию.

Межпредметные связи этой темы определяются кругом задач из других учебных предметов, для решения которых разрабатываются программы. В разделе развиваются некоторые представления о численных методах, формируемые в курсе математики.

Требования к знаниям и умениям. Учащиеся должны знать:

— название и назначение основных систем в базовом программном обеспечении школьной ЭВМ;

— названия и правила записи основных конструкций языка программирования; правила исполнения программ;

Учащиеся должны уметь:

— использовать основные системы базового программного обеспечения школьной ЭВМ, применяя «меню», запросы о «помощи», инструкции к пользованию;

— выражать конструкции алгоритмического языка средствами языка программирования;

— записать, отладить и исполнить простую программу на языке программирования; использовать стандартные функции и подпрограммы;

— организовать выдачу результатов исполнения программы на экран и принтер.

6. Решение задач на ЭВМ

6.1. Этапы решения задач на ЭВМ

Постановка задачи. Выделение предметной области, классификация объектов и их свойств. Построение математической модели. Определение аргументов и результатов. Построение алгоритма и разработка программы. Откладка и исполнение программы. Анализ результатов.

Примеры программирования и решения математических и физических задач.

6.2. Прикладное програмное обеспечение и его использование для решения задач.

Системы обработки текстовой информации. Структура текста. Операции работы с текстами. Понятие стандартного формата деловой документации. Хранение текстов во внешней памяти. Организация вывода текста на печать.

Системы машинной графики. Графические примитивы. Построение изображений с помощью графических примитивов.

Базы данных. Использование ЭВМ для хранения и систематизации данных. Картотеки. Понятие о базе данных как об автоматизированной картотеке. Основные операции над данными: создание картотеки, об-

новление данных, просмотр и поиск данных. Система картотек. Запросы к базе данных.

Системы работы с электронными таблицами. Понятие электронной таблицы. Развертка таблиц на экране. Основные операции работы с таблицами. Вычисления над таблицами.

Пакеты прикладных программ (ППП). Понятие о ППП как средстве автоматизации решения задач. Примеры ППП. Структура ППП. Организация взаимодействия с пользователем. Применение ППП для решения учебных задач.

6.3. Итоговый практикум. Коллективное решение учебно-производственных задач на ЭВМ.

Фронтальные практические работы: решение задач из школьных курсов; работа со структурами данных, с текстовым редактором, графическим редактором, базой данных, электронными таблицами и автоматическим решателем задач.

Практикум. 1. По разделу «Этапы решения задач на ЭВМ». 2. Работа с системой прикладного программного обеспечения (по выбору учителя). 3. Итоговый практикум. Коллективное решение задач с использованием готового программного обеспечения.

Межпредметные связи. Практически все содержание этой темы ориентируется на задачи различных школьных учебных предметов. Решение математических задач опирается на материал курса алгебры и основ математического анализа. Решение физических задач опирается на разделы механики, учение об электричестве и теплоте. Целесообразно ориентировать работу с автоматическим решателем задач на задачи из геометрии и тригонометрии, с системой обработки текстов — на литературную работу, с системой машинной графики — на задачи из черчения и геометрии, с системой электронных таблиц — на обработку экспериментальных данных из курсов физики или химии, с информационно-поисковой системой — на любые виды школьного делопроизводства.

Требования к знаниям и умениям. Учащиеся должны знать:

— названия и содержание основных этапов решения задач на ЭВМ;

— названия и назначение основных систем прикладного программного обеспечения.

Учащиеся должны уметь:

— зафиксировать предметную область и ее объекты, построить простую математическую модель, разработать и исполнить программу решения задачи на основе построенной модели, провести анализ результатов (для задач, аналогичных рассмотренным в учебнике);

— пользоваться текстовым редактором, организовать хранение текстов во внешней памяти и вывод их на печать в соответствии со стандартным форматом;

— пользоваться графическим редактором для построения несложных изображений;

— приводить примеры структуры данных, обращаться с запросами к базе данных, выполнять основные операции над данными;

— просмотреть электронную таблицу на экране, осуществлять основные операции над таблицами, выполнять простейшие вычисления над таблицей;

— применять учебные пакеты прикладных программ для решения задач из школьных учебных предметов.

7. ЭВМ в обществе

Краткая история развития информационно-вычислительной техники. Поколения ЭВМ. Представление о развитии элементной базы и программного обеспечения.

ЭВМ в производстве. Машины со встроенными микропроцессорами. Станки с числовым программным управлением. Промышленные роботы. Автоматизированные рабочие места.

ЭВМ в управлении и проектировании. Понятие об автоматизированных системах управления (АСУ) и системах автоматизации проектирования (САПР).

ЭВМ в науке, медицине, образовании и культуре. Компьютер в доме.

Роль ЭВМ в развитии современного общества. Электронизация общества. Задача автоматизации производства, повышения эффективности управления, экономии ресурсов и охраны окружающей среды. Роль компьютерной грамотности и информационной культуры в выработке новых трудовых навыков, в воспитании человека развитого социалистического общества.

Фронтальная практическая работа с учебными пакетами, демонстрирующими применения ЭВМ.

Межпредметные связи. Разделы, посвященные истории развития вычислительной техники (поколения ЭВМ, развитие элементной базы), опираются на знания, полученные из курсов физики.

Примеры применения ЭВМ в производстве, управлении, науке и других областях могут быть выбраны практически из всех школьных предметов.

Вопросы, связанные с ролью ЭВМ в современном обществе, тесно смыкаются с вопросами, рассматриваемыми в курсах истории и обществоведения.

Требования к знаниям и умениям. Учащиеся должны знать: основные этапы развития информационно-вычислительной техники, примерные характеристики поколений ЭВМ.

Учащиеся должны уметь:

приводить примеры применения ЭВМ в основных областях народного хозяйства, в повседневной деятельности человека; объяснять роль компьютерной грамотности и информационной культуры в общем и профессиональном образовании.

Примерный состав программного обеспечения курса.

1. Базовое программное обеспечение школьной ЭВМ (операционная система, файловая система, графическая система, текстовой редактор).
2. Языковая система программирования с библиотекой стандартных программ и системой отладки.
3. Клавиатурный тренажер.
4. Простой редактор текстов.
5. Простой графический редактор.
6. Учебный интерпретатор алгоритмического языка.
7. Учебная база данных.
8. Учебная система обработки электронных таблиц.
9. Демонстрационный пакет для предварительного знакомства с ЭВМ.
10. Семейство исполнителей с заданной системой команд и фиксированной обстановкой.
11. Библиотека вспомогательных алгоритмов.
12. Пакет программ, моделирующих работу ЭВМ и ее устройств.
13. Пакет моделирующих программ по темам из школьных курсов математики и физики.
14. Программная модель типовых структур данных.
15. Учебный пакет автоматического решателя задач.
16. Пакет программ управления учебным роботом.
17. Демонстрационный пакет по применениям ЭВМ.

НА КНИЖНОЙ ПОЛКЕ

Зеленко Г. В., Попов С. Н. **Проектирование микропроцессорных устройств на базе микропроцессора КР1810ВМ86.**— МЦНТИ, МНИИПУ, 1986.

Рассматриваются функциональные особенности и схемы включения БИС, работающих совместно с микропроцессором КР1810ВМ86: тактового генератора КР1810ГФ84, системного контроллера КР1810ВК88 и других.

Приводится пример структуры микропроцессорного устройства, на котором разбирается организация памяти и подсистемы ввода-вывода.

Заказы на издания МЦНТИ и МНИИПУ принимаются магазином № 93 «Книга-почтой» Москниги по адресу: 117168, Москва, В-168, ул. Кржижановского, 14, корп. 1.

МИКРОПРОЦЕССОРЫ

Тысячерукий как бог языческий
Твое величество —
Политехнический!
. . . .
Я ненавидел тебя в начале,
Как ты расстреливал меня молчанием.
Я шел как смертник в притихшем зале.
Политехнический, мы враждовали!
Ах, как я сыпался! Как шла на помощь
Записка искоркою электрической...
Политехнический, ты помнишь?

А. Вознесенский

С момента своего основания журнал «МП» проводит в Большом зале Политехнического музея центрального лектория общества «Знание» регулярные встречи авторского актива с читателями. Проводятся они с периодичностью 1—2 раза в месяц. Предмет обсуждения: последние разработки промышленности или схемотехнические находки талантливых «умельцев», отдельные программные решения или конкретный аппаратно-программный комплекс, вопросы применения или новые технические принципы. Неизменной остается лишь форма общения с аудиторией — авторы не столько рассказывают, сколько показывают свои результаты. На сцене обычно установлена действующая микропроцессорная аппаратура самого различного назначения: контроллер для управления цеховыми испытательными стендами, или станками с ЧПУ, «домашняя ЭВМ», или профессиональный персональный компьютер, «крейт КАМАК» или автомобильный «радиоцентр», звукосинтезатор для создания в диалоге с микроЭВМ музыкальных произведений или промышленные средства отладки встраиваемых микропроцессоров... Словом, это должен быть полезный в цехе, лаборатории или дома «компьютеризованный прибор» или система. На каждое очередное заседание выносится три — пять конкретных разработок.

Такие «встречи с журналом» регулярно собирают в «Политехническом» тысячную аудиторию специалистов Москвы и Подмосковья. Значительная часть участников приезжают из других городов (Киев, Харьков, Донецк, Ленинград, Новосибирск и др.). Представители промышленности находят здесь интересные для массового тиражирования результаты разработок вузовских или академических лабораторий, специалисты отраслевых НИИ и КБ приходят, чтобы следить за общей трассой развития идей и типовых разработок, а нередко и просто для того, чтобы установить личный контакт с заинтересовавшим их по выступлениям в журнале автором. Для той части читателей, которые по тем или иным причинам не могут приехать в «Политехнический», редакция периодически проводит выездные заседания «лектория МП». В конце 1985 начале 1986 года такие встречи были организованы в Обнинске, Дубне и других научных и промышленных центрах Подмосковья. В дальнейшем планируется постоянно расширять «географию» таких рабочих контактов журнала с читателями, в том числе, и далеко за пределы столичного региона.

Нет необходимости пояснять, как именно влияет такая «непечатная» сторона работы журнала на раз-



В ПОЛИТЕХНИЧЕСКОМ

витие «компьютерного потенциала» региона. С этой стороны, видимо, все понятно. Нередко, однако, приходится сталкиваться с другим вопросом: «Почему это нужно делать именно вам — всесоюзному журналу?» Наиболее часто приходится слышать этот вопрос от коллег из отраслевых журналов: «Что это Вы затеяли? Встречаемся конечно и мы с читателями, например, за юбилейным «круглым столом», посвященным десятилетию, двадцатипятилетию или пятидесятилетию какого-нибудь известного и заслуженного издания. Но не каждый же месяц! Контакт с читателем? Так есть же отдел писем...»

Итак, зачем нужен «Политехнический» журналу «МП»? Что дает столь интенсивное непосредственное общение с читателями самому журналу? Основной «трофей», который мы выносим из большого зала Политехнического музея, после напряженной многочасовой работы с аудиторией,— «мешки» с вопросами — записками, которые участники направляют ведущему и докладчикам. Анализ этой «почты реального времени» позволяет оперативно корректировать тематическую «розу ветров» журнала, так чтобы в наибольшей степени учитывать практические потребности специалистов народного хозяйства в актуальной информации по микропроцессорной технике. Например, только за последние полгода анализ «записок из Политехнического» позволил своевременно снять ряд готовых к набору, но, судя по реакции «зала» и редакционной почте, уже утративших пик актуальности тематических подборок, заменив их циклами по однокристальным ЭВМ, новым средствам отладки микропроцессорных систем, учебным циклом: «Учись работать с ПЗУ». Готовится цикл статей по программируемым логическим матрицам и т. д.

Таким образом, не реже, чем раз в месяц, тысячная аудитория «Политехнического» выражает со всей определенностью свое отношение и к тому, что уже было в журнале, и к тому, что хотелось бы нашим читателям там видеть. Члены редакционной коллегии, редсовета и редакции журнала имеют теперь возможность постоянно видеть перед глазами «колеблющуюся стрелку компаса» читательской аудитории.

Но почему же столь оперативная связь с читателем нужна именно журналу по микропроцессорной тематике, в то время когда без нее вполне благополучно обходятся, как известно, журналы более традиционного профиля? Дело в том, что в отличие от многих других, в том числе, и самых передовых областей науки и техники, бурно растущая область взаимосвязанных инженерных дисциплин, объединяемых понятием «информатика», изменяется не известными ранее в истории науки темпами. Два-три года — историческая эпоха. Концептуальная ось всех этих перемен проходит сегодня в области микропроцессорной техники. Поэтому и стиль работы единственного пока в стране журнала по этой тематике должен резко отличаться от стиля работы журналов по более традиционным направлениям техники. Есть основания предполагать, что журналу видимо в какой-то степени удалось найти приемлемую для значительной части читателей динамику развития тематического облика. Об этом позволяют судить не только письма читателей, ответы на анкеты и т. д., но и, видимо, более объективный показатель — темп роста тиража индивидуальной подписки. В первый год издания журнала у него было 7 тысяч подписчиков, второй — 20 тысяч, в 1986 году на журнал подписались — 50 тысяч читателей.

Открывая первый номер «МП» за 1984 год, А. П. Ершов отмечал: «Появление микропроцессорных средств радикально изменило характер использования и внедрения вычислительной техники. Главное здесь — массовость и необычайная широта применения. В недалеком будущем практически каждое рабочее место будет так или иначе затронуто переменами, связанными с возможностями микроэлектронной обработки информации. Редакционный совет, редакционная коллегия и сотрудники редакции хотят сделать журнал полезным каждому, кто связан с разработкой и применением микропроцессорных средств и систем. Задача состоит в том, чтобы реальные научно-технические достижения и удачные новшества сделать достоянием как можно большего числа разработчиков и потребителей. Мы хотим, чтобы журнал давал не только общую ориентацию в направлениях развития микроэлектронной вычислительной техники и ее компонент, но и был полезен в повседневной практической работе».



Фото Рикельмана А. Б.

Постояннодействующий лекторий «МП», который его участники иногда кратко называют «микропроцессоры в Политехническом», представляет собой попытку объединения усилий авторского актива, редакции и читателей с целью поиска наиболее эффективных путей решения поставленной перед новым журналом масштабной задачи ускорения процесса массовой компьютеризации народного хозяйства.

Г. Р. Громов, старший научный сотрудник
Научно-исследовательского вычислительного
центра АН СССР,
ведущий постояннодействующего лектория
журнала

От редакции. Кроме решения социально наиболее острой сегодня задачи — утоления «информационного голода» специалистов различных отраслей народного хозяйства по технике разработки и применения микропроцессорных средств автоматизации, журнал уделяет также значительное внимание и более общим вопросам развития информатики, ее роли в решении актуальных народнохозяйственных задач. Некоторые из этих вопросов обсуждались на тематическом заседании «круглого стола» журнала, который проводился в большом зале Политехнического музея в ноябре 1985 года. В последних номерах журнала за прошлый год мы знакомили читателей с некоторыми из выступлений за этим «круглым столом»: А. Г. Кушниренко, Н. А. Садовской, В. Ю. Романова. В первом номере 1986 года была опубликована стенограмма доклада А. П. Ершова. В третьем номере будут опубликованы материалы доклада Г. Р. Громова «Автоформализация профессиональных знаний».

Ниже мы публикуем репортаж из Политехнического, который был напечатан 4 декабря 1985 года в дубненской газете: «Дубна. Наука. Содружество. Прогресс». Репортаж этот представляет собой, разумеется, лишь личные впечатления двух из более, чем тысячи находившихся в зале участников встречи с журналом. Его авторы — научные сотрудники Лаборатории вычислительной техники и автоматики Объединенного института ядерных исследований в Дубне. Полный текст репортажа приводится здесь с разрешения редакции этой газеты.

Наша общая забота — школа

Сегодня потребность страны в кадрах, умеющих работать с вычислительными машинами, огромна. Самое непосредственное участие в подготовке будущих квалифицированных специалистов, в реализации реформы общеобразовательной школы, включая компьютеризацию обучения, принимают работники науки. Большой опыт в этом деле накоплен в ЛВТА ОИЯИ, на базе которой в течение нескольких лет овладели основами программирования и прошли практику сотни дубненских школьников. Многие сотрудники этой лаборатории преподают сейчас в школах новый предмет «Основы информатики и вычислительной техники».

Недавно в Москве состоялась встреча с редакцией журнала «Микропроцессорные средства и системы», уделяющего большое место на своих страницах проблемам компьютеризации. Ее участниками стали сотрудники ЛВТА А. А. Корнейчук и А. А. Расторгуев. Надеемся, их рассказ заинтересует и специалистов в этой области, и педагогов, и школьников.

«КРУГЛЫЙ СТОЛ» С ОСТРЫМИ УГЛАМИ

А. А. Корнейчук, А. А. Расторгуев

Ученики девятых классов вот уже три месяца изучают новую школьную дисциплину — «Основы информатики и вычислительной техники», а отношение к ней остается неоднозначным. Одни полагают, что введение информатики в школе преждевременно: нет ни квалифицированных кадров, ни настоящего учебника: не подготовлена техническая база. Все это может привести к дискредитации самой идеи всеобщей компьютерной грамотности. Другие считают, что ждать, когда появятся школьные компьютеры или терминальные классы, педагогические кадры, нельзя, что надо начинать, и начинать «круто», нет иной возможности сдвинуть дело с мертвой точки. Одни утверждают, что программирование должно идти от математики, как это произошло исторически, а в центре внимания должно быть построение алгоритма и доказательство его правильности. Другие доказывают, что обучать программированию без машины — утопия, обучать надо реальным языкам и на реальных машинах. Все эти вопросы так или иначе поднимались на недавней встрече с редакцией журнала «Микропроцессорные средства и системы» («МП»), организованной в Политехническом музее.

* * *

Первым на встрече было выступление академика **А. П. Ершова**, главного редактора «МП». Андрей Петрович приехать не смог, и выступление его было просмотрено по цветному телевизору. А. П. Ершов сравнил школьную информатику с кораблем, который мы должны выстроить общими усилиями, чтобы наши дети и внуки смогли войти на нем в третье тысячелетие. Корабль еще не достроен, но впереди него уже идет мутная волна. Кто-то не успел подготовиться, кто-то поспешил рапортовать, где-то на двухнедельных курсах учили чему угодно, но не информатике... Часто приходится слышать, что учить программированию без машин — все равно, что учить плавать в сухом бассейне, однако, сказал А. П. Ершов, мы твердо убеждены, что программирование должно идти от математики.

* * *

С проблемой школьной информатики тесно связана проблема компьютеризации школы. Это и материальная база для обучения программированию, и автома-

тизация труда учителя, использование ЭВМ при изучении других дисциплин. Опыт Новосибирска, где уже десять лет ведется программированное обучение на уроках математики и русского языка, начиная со второго класса, является уникальным. О том, как организован учебный процесс, о возникших проблемах рассказала **Н. А. Садовская** — педагог, психолог по специальности.

На сцене — ДВК-1, работающий в режиме терминала. Машина, к которой он подключен по обычной телефонной линии, находится в трех тысячах километрах в Новосибирске. Московская школьника Маша, за полчаса освоившая работу на терминале, решает задачи на сложение и другие арифметические действия, а также ставит запятые в предложениях, которые выдает машина. Машина отвечает ей «Молодец!» или поправляет, высвечивая на экране соответствующее правило грамматики. Опыт показывает, что школьники садятся за терминал с удовольствием. Им даже не так важна оценка учителя, им важно, чтобы их похвалила машина.

Н. А. Садовская подробно остановилась и на некоторых закономерностях, установленных в процессе программированного обучения, а также на трудностях, к которым приводит использование ЭВМ в школе. Например, нагрузка на педагогов возрастает настолько, что это может привести к эмоциональному срыву. Выявилось также резкое различие в способностях учеников, точнее, в их реакции: «шустрики» по скорости порой отличаются от «мямликов» на порядок.

* * *

После темпераментного выступления Н. А. Садовской состоялся доклад **А. Г. Кушниренко**, посвященный преподаванию программирования в вузе. Как истинный математик Кушниренко начал «от противного»: я преподаю на мехмате МГУ уже 15 лет, и за это время хорошо понял, как не надо учить программированию. Во-первых, не надо учить справочным сведениям, особенно из операционной системы ЕС... Выпад против ЕС был встречен рукоплесканиями зала. Большой зал Политехнического музея, вмещающий около тысячи человек, в этот раз был переполнен, поэтому поддержка получилась внушительной. Если и были там те, кто думал иначе, их голоса потонули в общем гуле одобрения. Что представляет собой сейчас, продолжал докладчик, изучение информатики в вузе? Это нечто случайно сложившееся, не имеющее отношения ни к потребностям учебного процесса, ни к потребностям будущих профессий студентов. Обучение проходит на устаревшей вычислительной технике, с очень устаревшим программным обеспечением. Даются некоторые сведения о ЕСовской аппаратуре и численные методы, и решаются две-три задачи. Такое же положение во многих других вузах.

* * *

Три-четыре года назад перед разработчиками ЭВМ была поставлена задача государственной важности: в сжатые сроки создать отечественные профессиональные и учебные персональные компьютеры. Опыт показал, что так же, как и за рубежом, самодеятельные коллективы разработчиков вполне успешно конкурируют с хорошо финансируемыми солидными «фирмами». **Персональный компьютер «Ириша»** был представлен группой разработчиков из трех человек с химического факультета МГУ. Программное обеспечение для нее «делали» в Московском институте проблем информатики. «Ириша» стояла тут же, в президиуме, и выводила на экран телевизора постоянно меняющуюся объемную картинку. У нее 64 килобайта оперативной памяти, гибкие диски, сенсорная клавиатура, она свободно подключается к бытовому телевизору и кассетному магнитофону По вычислительным возможностям компьютер немного уступает ДВК-1. Сейчас «Ириша» демонстрируется на ВДНХ в павильоне «Образование». Там же можно увидеть «Микрошу», «Агат», «Искру». «Ириша» может претендовать на роль школьного компьютера, она также может использоваться в АСНИ — автоматизированных системах научных исследований. Серийное ее производство еще не начато, а стоимость опытных образцов около 1000 рублей.

* * *

Примерно в то же самое время, когда вовсю стали говорить о компьютерной грамотности, заговорили и о «кризисе прикладного программирования». Математические модели, «наработанные» за 300 лет, покрывают сравнительно узкую область явлений природы, технологических процессов и человеческой деятельности. Когда программист-профессионал хочет постичь для себя новую область, где нет математической модели, он идет к другому специалисту и пытается получить от него профессиональные знания, чтобы положить их на язык программирования, но наталкивается на трудности почти непреодолимые. Приблизительно только 20 процентов из того, что мы знаем, можно объяснить своему коллеге, на порядок меньше — человеку непосвященному и еще на порядок меньше — записать в формальном виде, исключающем различное толкование. Однако когда специалист в своей области (непрофессиональный программист) садится за терминал, происходит таинство: минуя строгую постановку задачи, блок-схемы и алгоритмы, он начинает писать программу. Таково вкратце содержание доклада **Г. Р. Громова**, ответственного секретаря «МП» и «главного распорядителя» этой встречи. Более подробно с этими аспектами прикладного программирования можно ознакомиться в книге Громова «Национальные информационные ресурсы».

* * *

Одна из основных проблем в программировании — поиск ошибок, процесс порой долгий и мучительный, потому что, как правило, известно, что ошибка есть, но неизвестно, где она находится. Радикальный способ борьбы с ними — с самого начала писать безошибочные программы. В качестве инструментария предлагаются методы структурного программирования, пошаговая детализация алгоритма, описание постановки задачи и доказательства правильности программы. Так обучают программистов в Московском институте электронного машиностроения. С докладом об опыте работы со студентами МИЭМ выступил преподаватель института **В. А. Каймин**. После его выступления счел нужным высказаться А. Г. Кушниренко: «Нам нужны не программы без ошибок, нам нужны работающие программы. Ошибки в программах — это не теоретическая проблема, а чисто экономическая...».

* * *

Завершил встречу «за круглым столом» доклад **М. Я. Каца**, одного из авторов персонального компьютера «Искра». Работа эта выдвинута на соискание Государственной премии. Серийное производство «Искры» начато в 1981 году, и в настоящее время в самых разных НИИ и отраслях народного хозяйства работают тысячи этих машин.

Несмотря на жесткий регламент, встреча длилась более четырех часов...

БЛИЖАЙШИЕ СЕМИНАРЫ ЖУРНАЛА:

13 мая 1986 года «Компьютерные игры», 27 мая — «Диалог человек — ЭВМ: психологические аспекты», 24 июня — «Тенденции развития мировой индустрии ЭВМ». Место проведения — Большой зал Политехнического музея. Начало в 12 ч.

Справки о семинарах можно получить по телефону 923-00-19 у методиста Политехнического музея Ермолаевой Татьяны Юрьевны.

Рефераты статей

УДК 681.322.068
Кобылинский А. В., Береза А. В., Сабадаш Н. Г., Тесленко А. К., Темченко В. А. **Система команд микропроцессора КМ1810ВМ86.** — Микропроцессорные средства и системы, 1986, № 2, с. 3.

Приведены краткая характеристика программно-доступных элементов, форматы команд и данных, система команд и алгоритмы их выполнения.

UDC 681.322.068
Kobylinsky A. V., Berjoza A. V., Sabadash N. G., Teslenko A. K., Telichenko V. A. **KM1810VM86 Instruction Set.**— Microprocessor Devices and Systems, 1986, N 2, p. 3.

The characteristics are given of KM1810VM86 microprocessor instruction set.

УДК 621.349.77 : 681.325.5
Белоус А. И., Горовой В. В., Прибыльский А. В., Силин А. В. **Интерфейсные интегральные схемы серии К583.** — Микропроцессорные средства и системы, 1986, № 2, с. 9.

Приведены структурные схемы, временные диаграммы работы, электрические характеристики интерфейсных интегральных схем серии К583: магистрального приемопередатчика с памятью К583ВА1, магистрального приемопередатчика К583ВА2, универсального 8-разрядного коммутатора магистралей К583ВА3, универсального 8-разрядного магистрального коммутатора К583ВА4.

UDC 621.3.049.77:681.325.5
Belous A. I., Gorovoi V. V., Pribylsky A. V., Silin A. V. **The Series of K583 Intergrated Circuits.**— Milcroprecessor Devices and Systems, 1986, N 2, p. 9.

The paper presents structural schemes, timing diagrams and electrical characteristics of K583 IC set: data bus transceivers and commutators K583VA1, VA2, VA3, VA4.

УДК 681.325.5:621.3.049.77
Калошкин Э. П., Белоус А. И., Палиенко В. В., Подрубный О. В., Журба В. М. **БИС универсального процессорного элемента К1815ВФ1.**— Микропроцессорные средства и системы, 1986, № 2, с. 15.

Приведены структурная схема, временная диаграмма работы, описание функциональных возможностей и работы БИС универсального процессорного элемента. Микросхема осуществляет прием двоичных операндов в последовательно-параллельном (по два разряда в каждом такте) прямом коде, перемножает принятые операнды и суммирует полученные произведения (или только перемножает принятые операнды) и выдает результат этих операций в последовательно-параллельном дополнительном коде на выходные магистрали данных. Темп приема, обработки и выдачи данных достигает 8,4 МГц (период основного синхросигнала — 118 нс). По электрическим параметрам БИС полностью совместима с быстродействующими микросхемами ТТЛ и ТТЛШ серий.

UDC 681.325.5:621.3.049.77
Kaloshkin E. P., Belous A. I., Podrubny O. V., Palienko V. V., Zhurba V. M. **K1815VF1 Universal Processor Element.**— Microprocessor Devices and Systems, 1986, N 2, p. 15.

The general structure, timing diagram and description of functional features of universal processor element are given in the paper. The device miltiplies binary operands in serial-parallel code and produces the sum of multiplication results at the output. The frequency rate is close to 8,4 MHz. The device is TTL — compatible with other LSIs.

УДК 621.3.049.770:681.325.3
Милованов А. И., Силин А. В., Сахаров А. М., Яковлев Ю. В. **Быстродействующий сумматор последовательных чисел К1815ИМ1.** — Микропроцессорные средства и системы, 1986, № 2. с. 18.

Приведены структурная схема, временная диаграмма работы, электрическая схема, описана работа микросхемы быстродействующего сумматора последовательных чисел К1815ИМ1. Микросхема предназначена для суммирования восьми двоичных чисел неограниченной разрядности, представленных в последовательно-параллельном дополнительном коде. По электрическим параметрам микросхема совместима с ИМС серий К155, К133, К586, К584, К598, К1533. Микросхема осуществляет суммирование нескольких последовательно-параллельных чисел.

UDC 621.3.049.770:681.325.5
Milovanov A. I., Silin A. V., Sakharov A. M., Jakovlev Yu. V. **K18151M1 High-Speed Adder.**— Microprocessor Devices and Systems, 1986, N 2, p. 18.

The general structure, timing and electrical diagrams are described for K18151M1 high-speed serial adder. The chip can perform addition of eight binary numbers of unlimited length in serial-parallel complementary code. The chip is electrically compatible with LSIs K155, K133, K586, K598, K1533.

УДК 621.372.542:681.31.002:621.382.8
Бобков В. А., Сухопаров А. И. Гузик В. Ф., Каляев А. В., Попов Ю. П., Станишевский О. Б. **Микропроцессор К1815ВФ3 для цифровой обработки сигналов.** — Микропроцессорные средства и системы, 1986, № 2, с. 21.

Приведены основные электрические характеристики и структура однокристального микропроцессора, предназначенного для выполнения таких операций, как быстрое преобразование Фурье, умножение и сложение комплексных чисел, перекурсивный и рекурсивный фильтры, полином и т. д.

UDC 621.542:681.31.002:621.382.8
Bobkov V. A., Sukhoparov A. I., Guzik V. F., Kalyaev A. V., Popov Yu. P., Stanishevsky O. B. **K1815VF3 Microprocessor for Digital Signal Processing.**— Microprocessor Devices and Systems, 1986, N 2, p. 21.

The paper presents electrical characteristics and general structure of monochrystal microprocessor intended for such operations as fast Fourier transformation, addition and multiplication of complex numbers, calculation of recursive filters, polynoms etc.

УДК 681.322.1

Тилинин Д. А. **Персональная ЭВМ «Океан 240».**— Микропроцессорные средства и системы, 1986, № 2, с. 24.

В краткой форме описана персональная ЭВМ «Океан 240», разработанная на основе микропроцессорного комплекта БИС серии К580. Все устройства ПЭВМ, объединенные системной шиной, расположены на одной печатной плате. Простота сопряжения с нестандартной аппаратурой обеспечивается набором программируемых устройств параллельного и последовательного обмена.

UDC 681.322.1

Timenin D. A. **"Okean 240" Personal Computer.**— Microprocessor Devices and Systems, 1986, N 2, p. 24.

"Okean 240" singleboarded personal computer is described which is developed on the basis of K580 microprocessor. The simplicity of interface with nonstandard apparatus is provided by means of programmable serial and parallel controllers connected to the common system bus.

УДК 681.3—181.1.06

Куклин В. В., Калинин Н. М., Бобров Ю. А. **Отладочный модуль на базе однокристального микропроцессора К1801ВМ1.** — Микропроцессорные средства и системы, 1986, № 2, с. 33.

Рассмотрен отладочный модуль, имеющий программу монитор с широким набором функций, диагностическими сообщениями об ошибках и возможностью создания библиотеки подпрограмм. Приведены функциональные и принципиальные схемы устройства, распечатка программы монитор в машинных кодах.

UDC 681.3-181.1.06

Kuklin V. V., Kalinin N. M., Bobrov Ju. A. **Debugging Module on a Monocrystal Microprocessor K1801VM1.**— Micrporocessor Devices and Systems, 1986, N 2, p. 33.

The authors describe the multifunctional debugging device with appropriate software. The sytem provides error messages, creating the library of functional subprograms and other debugging means. Logic diagrams and machine code program listings are presented.

УДК 681.326—181.4

Фельдман В. М. **Проверка и ремонт микроЭВМ в условиях объекта эксплуатации.** — Микропроцессорные средства и системы, 1986, № 2, с. 37.

Предлагается методика восстановления работоспособности систем управления на базе микроЭВМ персоналом без специальной подготовки по вычислительной технике.

UDC 681.326—181.4

Feldman V. M. **Microcomputer Testing and Rapair in the Field Environment.**— Microprocessor Devices and Systems, 1986, N 2, p. 37.

The author discusses the methods of technical support for microcomputer-based systems when a personnel does not have special training in computers.

УДК 681.327.023 : 621.391.64

Раденко М. Е., Сеньков В. К. **Реализация волоконно-оптического интерфейса для микропроцессорных систем.** — Микропроцессорные средства и системы, 1986, № 2, с. 41.

Предложено устройство для «удлинения» шин системного интерфейса микроЭВМ на базе БИС серии К580 с помощью волоконно-оптических компонентов. Приводятся структурные схемы и временные диаграммы работы волоконно-оптических мультиплексоров, сопряженных с распределенной микропроцессорной системой управления.

UDC 681.327.023:621.391.64

Radenko M. E., Senkov V. K. **Realization of Realization of Fibre-Optical Interface for Microprocessor Systems.**— Microprocessor Devices and Systems, 1986, N 2, p. 41.

The paper describes fibre-optical device for extending microcomputer systems interface bus of LSI K580. Structural diagrams and the timing characteristics of fibre-optical multiplexors for a distributed microprocessor control system are discussed.

УДК 681.7.068:681.327.21/.22

Баронец В. Д., Бойцов Р. Н., Воробьев А. Г., Забегалова Г. И. **Устройства связи микроЭВМ с объектами управления по волоконно-оптическим линиям.**— Микропроцессорные средства и системы, 1986, № 2, с. 43.

Рассматривается функционирование модуля приемопередающего «Электроника МС 4605» и модулей интерфейсных «Электроника МС 8801», «Электроника МС 8802» для микроЭВМ «Электроника 60М», «Электроника НЦ-80 01Д», «Электроника С5-41». Приводятся общие виды, принципиальная и структурная схемы, технические характеристики модулей. Рекомендуется применять модули для создания микроУВК различной архитектуры с использованием помехозащищенных и экономичных волоконно-оптических линий связи для автоматического управления производственными объектами.

UDC 681.7.068:327.21/22

Baronets V. D., Boitsov R. N., Vorobjov A. G., Zabegalova G. I. **Fiber-Optical Connection of Microcomputer with Controlled Objects.**— Microprocessor Devices and Systems, 1986, N 2, p. 43.

The paper presents the principles of "Electronica MS 4605", "Electronica MS 8801", "Electronica MS 8802" functioning and their technical specifications. Recommendations are given for building noise-tolerant communication lines between microcomputers and controlled objects on the basis of fibre-optical communication lines.

УДК 681.322.1

Барышников В. Н., Быстров В. П., Воронов М. А., Папачев Ф. И., Романов В. Ю. **Модуль процессора персональной ЭВМ «Ириша».**— Микропроцессорные средства и системы, 1986, № 2, с. 52.

Приводится подробное описание собственно процессора, последовательного и параллельного интерфейсов, интерфейса клавиатуры, таймера, синтезатора звука, игрового адаптера.

Перечислен состав программного обеспечения, даны основные инструкции по наладке модуля. Представлены принципиальная и монтажная схемы модуля, топология печатной платы.

UDC 681.322.1

Baryshnikov V. N., Bystrov V. P., Voronov M. A., Panochev F. I., Romanov V. Yu. **Processor for "IRISHA" Personal Computer.**— Microprocessor Devices and Systems, 1986, N 2, p. 52.

The paper presents a detailed description of "IRISHA"'s microprocessor, serial and parallel interfaces, keyboard interface, timer, sound generator and game adapter. Software is described. Debugging procedures, motherboard layout and logical diagrams are presented.

УДК 681.32

Лукьянов Д. А. **«Электроника 256К» — эмулятор диска для комплексов на основе микроЭВМ «Электроника 60» и ДВК.** — «Микропроцессорные средства и системы», 1986, № 2, с. 62.

Рассмотрены технические и программные средства, необходимые для включения серийных блоков массовой памяти емкостью до 2 Мбайт и более в качестве эмулятора устройства с файловой структурой прямого доступа в операционную систему РАФОС.

Каждый блок ОЗУ эквивалентен при этом диску емкостью 1010 информационных блоков, а время доступа к хранимой информации примерно в 100 раз выше, чем в накопителях на гибких магнитных дисках. Применение эмулятора в небольших персональных измерительно-вычислительных комплексах позволяет пользоваться компиляторами Ратфор/Паген/Фортран 4, Паскаль 2, Модула 2 и заметно увеличить скорость трансляции работы прикладных оверлейных программ.

UDC 681.32

Lukjanov D. A. **"Electronica 256K".— Disk Emulafor "Electronica 60" and DVK.**— Microcomputers.— Microprocessor Devices and Systems, 1986, N 2, p. 62.

The author describes hardware and software components of RAM disc emulator for OS RAFOS. The memory capacity can be as high as 2 MB.

Each RAM block substitutes the floppy disk with capacity of 1010 information blocks with the access time approximately 100 times shorter than for floppy disks. Application of emulator allows speeding up program compilation for such languages as Ratfor, Pagen, Fortran, Pascal 2, Modula 2.

УДК 681.326.3

Щелкунов Н. Н., Дианов А. П. **Процедуры программирования логических матриц.** — Микропроцессорные средства и системы, 1986, № 2, с. 71.

Приведены структура и основные технические характеристики электрически программируемых логических матриц типа КР556РТ1 и КР556РТ2. Рассмотрены принципы и алгоритмы их программирования, контроля и редактирования.

UDC 681.326.3

Schelkunov N. N., Dianov A. P. **Logic Array Programming.**— Microprocessor Devices and Systems, 1986, N 2, p. 71.

The structure and main technical features are presented for programmable logical arrays KR556RT1 and KR556RT2. The programming principles and algorithms are presented.

УДК 681.326.3

Дианов А. П., Щелкунов Н. Н. **Технические средства программирования логических схем.** — Микропроцессорные средства и системы, 1986, № 2, с. 77.

Предложена программная модель и схема модуля программирования ПЛМ и ПЗУ. Модуль подключается к стандартной внутрисистемной магистрали типа И41 и совместим с универсальной программой прожига UPP версии 3.7, входящей в состав программного обеспечения микроЭВМ СМ 1800.

UDC 681.326.3

Dianov A. P., Schelkunov N. N. **Logic Array Programmers.**— Microprocessor Devices and Systems, 1986, N 2, p. 77.

The author describes software and hardware components of the system intended for programming Logic Arrays and ROMs. The module is compatible with the standard system interface I41 and PP programmer, version 3.7. It is intended for operation on SM-1800 computer.

УДК 681.326.3

Щербаков О. А. **Особенности применения ПЛМ в микропроцессорных системах.**— Микропроцессорные средства и системы, 1986, № 2, с. 80.

Рассматриваются особенности реализации электрических схем с использованием ПЛМ, даются правила и примеры заполнения таблиц программирования для ввода информации в ЭВМ. Приведены практические примеры построения дешифратора, Т-триггера и коммутатора периферийных устройств на ПЛМ.

UDC 681.326.3

Scherbakov O. A. **On Application of Programmable Logic Arrays in Microcomputer System.**— Microprocessor Devices and Systems, 1986, N 2, p. 80.

The peculiarities of logic diagrams are considered for application of Programmable Logic Arrays. The author describes the rules and examples of filling programming tables. Some practical examples are considered for building addressing scheme, T-trigger and peripheral commutator with PLA.

ИНФОРМАЦИОННАЯ СИСТЕМА ДЛЯ РУКОВОДСТВА НА ОСНОВЕ ПЕРСОНАЛЬНОЙ ЭВМ

Т. В. Ускова

Информационная система для руководства должна обеспечивать ввод и корректировку административной информации о структуре, кадровом составе и деловой активности учреждения, а также возможность удобного и быстрого доступа к этой информации.

Типичный пользователь рассматриваемой системы — это специалист, имеющий дело с разнообразными документами, — руководитель, референт, секретарь и другие сотрудники административно-управленческого аппарата предприятий и учреждений [1, 2].

Система может содержать следующие разделы:

1. Структура учреждения — перечень подразделений, их руководителей, телефоны, адреса, численный состав.
2. Состав подразделений — названия, списки сотрудников, их анкетные данные, служебные характеристики.
3. Специализация — основные направления (темы) работ, исполнители, сроки, связь с общегосударственными программами.
4. Используемое оборудование.

Основные режимы работы: А — ввод и корректировка административной информации, осуществляемые помощниками руководителя, секретарями; Б — обращение к системе со стороны руководящих работников.

В режиме А пользователь применяет привычную повседневную методику работы с информацией — выбор нужного документа или его части, внесение в него новых данных или исправлений. Информация хранится в двух основных формах — в текстовых файлах и в таблицах. Текстовые файлы создаются и модифицируются текстовым процессором. При этом используются обычные приемы работы с текстами — впечатывание новых символов, стирание, замена фрагментов и др. При работе с таблицами сначала строятся сами таблицы, состоящие из произвольного числа именованных полей, а затем они заполняются текстовыми или числовыми данными. Число строк в таблице не ограничено, но на экране, в соответствующем дисплейном окне всегда видно лишь несколько строк (окно как бы скользит вдоль таблицы, открывая доступ к новым строкам). Каждая ячейка таблицы может содержать число или текст длиной в одну строку. При просмотре таблицы видна лишь начальная часть такой строки, определяемая шириной столбца; но имеется отдельное дисплейное окно, в котором индивидуальные строки можно увидеть полностью, изменить или удалить их.

В режиме Б на экран сначала выводится тексто-графическое изображение, отдельные фрагменты которого отображают организационную структуру учреждения и/или основные информационные разделы. Это изображение играет роль управляющего меню, в котором пользователь может выбрать необходимый вид работы (документ) для просмотра (рис. 1). Интересующий его пункт меню выбирается подвижным курсором, перемещаемым по экрану манипулятором «мышь» или под управлением специальных клавиш на клавиатуре.

Меню может указывать на конкретные виды работы или на другие (подчиненные) меню. В зависимости от характера информации, доступ к которой обеспечивает система, исходное изображение, играющее роль управляющего меню, может быть картой-схемой страны, района или отдельного предприятия. Это может быть просто условная картинка, на которой различными визуальными средствами выделяются позиции, отображающие функциональные возможности системы (рис. 2). Вся совокупность меню образует «схему диалога», обрабатываемую специальной программой — Адаптивным диалоговым монитором [3]. После выбора в меню вида работы или необходимого документа система переходит на следующий уровень: выводит на экран текст документа или таблицу с требуемой информацией.



Рис. 1

Пользователь выбирает один, два или несколько интересующих его разделов информации, и система выводит соответствующие данные в несколько дисплейных окон [4]. При «прокрутке» одного окна содержимое соседних окон не меняется. Нажав на специальные клавиши, можно перейти от одного окна к другому.

Для работы с системой не требуется владение специальными знаниями, фактически не нужно отдельного обучения методам работы. Описанная система обеспечивает удобный и быстрый доступ пользователям к информации разного рода. Соблюдается принцип «выбирай и указывай», характерный для современного стиля работы на персональных компьютерах.

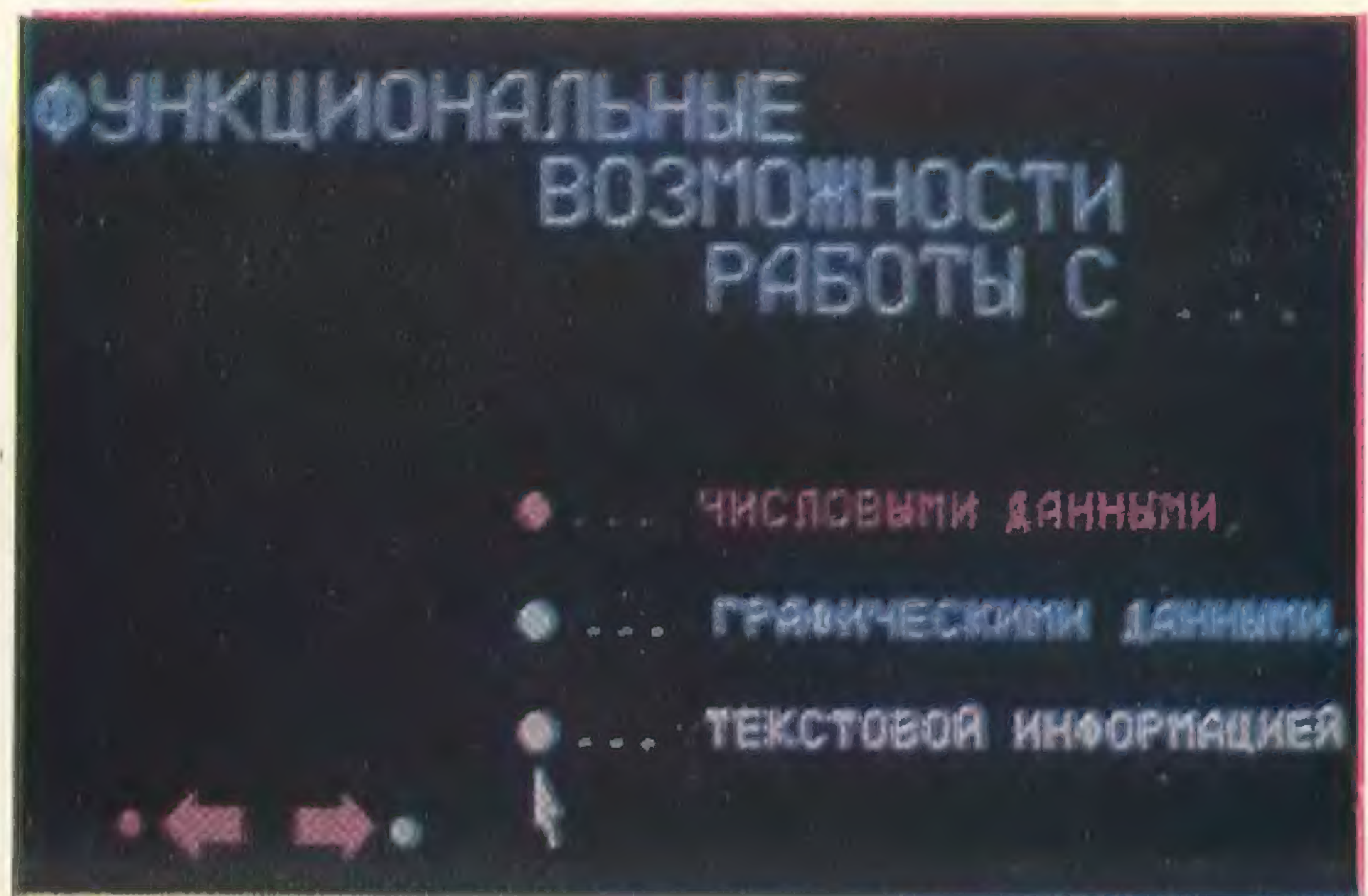


Рис. 2

ЛИТЕРАТУРА

1. Ершов А. П. Автоматизация работы служащих.— В кн.: Персональные ЭВМ в задачах информатики.— Новосибирск, 1984.

2. Брябрин В. М. Интеллектуальный интерфейс на основе персональной ЭВМ.— Микропроцессорные средства и системы, 1984, № 3, с. 38.

3. Ускова Т. В. Реализация информационной системы на персональном компьютере.— Proc. of the 3rd Int. Meeting of Young Computer Scientists.— Budapest: Computer and Automation Institute of the Hungarian Academy of Sciences, 1984.

4. Борковский А. Б. Многооконное текстовое взаимодействие с персональной ЭВМ.— Микропроцессорные средства и системы, 1984, № 4, с. 47.



МИКРОЭВМ ТИПА VT 16

МикроЭВМ типа VT 16, функционирующая в двух режимах работы, относится к классу профессиональных персональных ЭВМ. Два независимых режима работы обеспечивают выполнение программ, написанных для 8-разрядной и 16-разрядной микроЭВМ, и таким образом в 8-разрядном режиме работы без изменения могут исполняться программы, написанные для операционной системы UPM (соответствующей CP/M) для VPPC и VT 20/A. VT 16 выпускается в двух вариантах:
— с двумя встроенными мини-НГМД, емкостью по 1М байт;
— с одним мини-НГМД (1М байт) и с одним НМД типа WINCHESTER (5 или 10М байт).



ТЕХНИЧЕСКИЕ ДАННЫЕ

● оперативная память в базовой конфигурации — 256К байт RAM
● монитор (экран) 14 дюймов с матовым покрытием; отображение 24×80 символов; графическое действие по принципу BIT — MAPPING; 32К байт RAM
● интерфейс для передачи данных:
— МККТТ V.24
асинхронный / синхронный (ориентированный на бит и на байт)
скорость передачи данных 50—38400 бод
— МКТТ V.24/LAN.
со стыковкой к локальной сети
● интерфейс к АЦПУ:
параллельный, совместимый с АЦПУ типа CENTRONICS
● матричное печатающее устройство VT 21200.

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

● в режиме работы с 8-разрядным процессором:
операционная система UPM (соответствующая CP/M)
COBLAB
UPM — BASIC

● в режиме работы с 16-разрядным процессором:
операционная система, совместимая с MS DOS
FORTRAN
BASIC
операционная система, совместимая с CP/M 86.

